



TUGAS AKHIR - TE141599

IMPLEMENTASI RFID TAGS PADA SISTEM KONTROL PINTU GESER OTOMATIS BERBASIS MIKROKONTROLER

Heriyanto
NRP 2212105041

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Ir. Tasripan, MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE141599

***IMPLEMENTATION OF RFID TAGS IN AUTOMATIC
SLIDING DOORS CONTROL SYSTEM BASED ON
MICROCONTROLLER***

Heriyanto
NRP 2212105041

Supervisor
Dr. Muhammad Rivai, ST., MT.
Ir. Tasripan, MT.

*ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015*

**IMPLEMENTASI RFID TAGS PADA SISTEM KONTROL
PINTU GESER OTOMATIS BERBASIS
MIKROKONTROLER**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada
Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

Menyetujui:

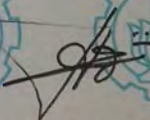
Dosen Pembimbing I,



Dr. Muhammad Rivai, ST., MT.

NIP: 196904261994031003

Dosen Pembimbing II,



Ir. Tasripan, MT.

NIP: 196204181990031004



Implementasi RFID Tags Pada Sistem Kontrol Pintu Geser Otomatis Berbasis Mikrokontroler

Nama : Heriyanto
Pembimbing : Dr. Muhammad Rivai, ST., MT.
Ir. Tasripan, MT.

ABSTRAK

Kesibukan dan mobilitas yang dimiliki manusia membuat manusia ingin mendapatkan kemudahan dalam kegiatannya. Dalam kehidupan sehari-hari penggunaan pintu geser masih dibuka ataupun ditutup secara manual. Tugas akhir ini merancang sistem pengendali pintu geser otomatis berdasarkan pembacaan identitas (ID) yang dimiliki pada *Radio Frequency Identification* (RFID) menggunakan mikrokontroler dengan tujuan untuk menggantikan pekerjaan secara manual menjadi otomatis. Penggerak pintu geser menggunakan motor DC yang dikontrol menggunakan kontroler PID yang ditanamkan pada mikrokontroler. Untuk mendapatkan konstanta kontroler PID dilakukan dengan menggunakan metode Ziegler-Nichols untuk memperbaiki kinerja dari motor DC. Hasil Pengujian kontroler PID mampu memperkecil error rata-rata dari set point yang semula mencapai 14,8 % menjadi 3,6% setelah diberikan kontrol loop tertutup dengan PID. Penggunaan RFID Tags dapat menggantikan penggunaan kunci atau *remote* sebagai pengendali kondisi pintu geser secara otomatis.

Kata kunci: Kontroler PID, Pintu Geser, RFID.

--Halaman ini sengaja dikosongkan--

Implementation of RFID Tags in Automatic Sliding Door Control System Based On Microcontroller

Name : Heriyanto
Supervisor 1 : Dr. Muhammad Rivai, ST., MT.
Supervisor 2 : Ir. Tasripan, MT.

ABSTRACT

Activity and mobility of human being make people want to get the ease of operation. In everyday the use of sliding doors is opened or closed manually. The final project is to design an automatic sliding door control system based on the reading of the Identity (ID) held on Radio Frequency Identification (RFID) using microcontroller to replace manual work become automatic. The sliding door was driven by DC motor controlled with PID method embedded in the microcontroller. PID Controller constants obtained with Ziegler-Nichols method to improve the performance of DC motor. Test Result of PID controller is able to minimize the average error of the set point which originally reached 14.8% to 3.6% after controlled. The use of RFID tags can replace the use of a key or a remote control automatic sliding door condition.

Keywords: *PID controller, RFID, Sliding Doors.*

--Halaman ini sengaja dikosongkan--

KATA PENGANTAR

Segala puji syukur penulis panjatkan atas kehadiran Allah SWT yang selalu memberikan rahmat serta hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Pada Kesempatan ini penulis ingin mengucapkan rasa syukur kehadiran Allah SWT yang telah memberikan petunjuk, kemudahan, dan kemurahan-Nya serta tidak lupa ucapan terimakasih yang sebesar-besarnya kepada beberapa pihak yang telah memberikan dukungan selama proses pengerjaan tugas akhir ini, antara lain:

1. Allah pemilik alam semesta dan Rosulnya .
2. Ibu tercinta yang selalu memberi semangat, doa dan kasih sayangnya.
3. Bapak Dr. Muhammad Rivai, ST., MT. dan Ir. Tasripan, MT. sebagai pembimbing yang telah membantu dan membimbing hingga laporan ini terselesaikan.
4. Bapak dan Ibu penguji Tugas Akhir yang telah memberi masukan kepada penulis untuk menyusun buku ini menjadi lebih baik.
5. Teman-teman Teknik Elektro ITS yang telah berbagi suka dan duka selama ini.
6. Semua pihak yang turut membantu pengerjaan Tugas Akhir ini yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini dikarenakan keterbatasan kemampuan penulis. Penulis mengharapkan berbagai masukan dan kritik dari berbagai pihak agar lebih baik di masa yang akan datang. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan ilmu pengetahuan demi kemajuan Indonesia menuju negeri berteknologi mandiri.

Surabaya, Januari 2015

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi	2
1.6 Sistematika Penulisan	3
1.7 Relevansi	3
BAB II LANDASAN TEORI	5
2.1 Motor Arus Searah (DC).....	5
2.2 <i>Radio Frequency Identification</i> (RFID).....	6
2.2.1 RFID Tags.....	7
2.2.2 RFID Reader	8
2.3 Mikrokontroler	8
2.3.1 Mikrokontroler AT-Mega32.....	10
2.4 Sensor Jarak	12
2.4.1 Sensor Ultrasonic	12
2.5 Sensor Kecepatan	14
2.5 Kontroler PID.....	15
2.5.1 Kontroler Proporsional (P).....	15
2.5.2 Kontroler Integral (I)	16
2.5.1 Kontroler Derivatif (D).....	16
2.6 <i>Pulse Width Modulation</i> (PWM)	16
BAB III PERANCANGAN SISTEM	21
3.1 Arsitektur Sistem	21
3.2 Perancangan <i>Prototype</i> Pintu Geser	23

3.3 Perancangan Perangkat Keras (<i>Hardware</i>)	24
3.3.1 Perancangan Mekanik <i>Plant</i>	26
3.3.2 Perancangan Rangkaian Mikrokontroler	26
3.3.3 Perancangan Rangkaian <i>Driver</i> Motor DC	29
3.3.4 Perancangan Sensor Kecepatan <i>Rotary Encoder</i>	31
3.3.4 Perancangan Sensor Jarak	32
3.3.4 Perancangan RFID.....	34
3.4 Perancangan Perangkat Lunak (<i>Software</i>).....	36
3.4.1 Perancangan Pembacaan ID Pada RFID Tags.....	37
3.4.2 Perancangan <i>Software</i> Sensor Jarak.....	38
3.4.3 Perancangan <i>Software</i> Sensor Kecepatan.....	38
3.4.3 Perancangan <i>Software</i> Pengendali <i>Driver</i> Motor DC	39
3.5 Identifikasi <i>Plant</i>	40
3.5.1 Metode Identifikasi.....	40
3.6 Perancangan Kontroler	42
BAB IV HASIL PENGUJIAN DAN ANALISA	47
4.1 Pengujian Parsial	48
4.1.1 Pengujian Rangkaian Mikrokontroler ATmega-32...	48
4.1.2 Pengujian PWM Dengan <i>Driver</i> Motor DC	51
4.1.3 Pengujian Sensor Jarak Ultrasonic HC-SRF04	53
4.1.4 Pengujian Pembacaan Kecepatan Motor	55
4.1.5 Pengujian Pembacaan RFID Tags	57
4.1.6 Pengujian Kontroler PID Yang Digunakan	59
4.2 Pengujian Integrasi	62
BAB V PENUTUP	65
5.1 Kesimpulan	65
5.2 Saran	65
DAFTAR PUSTAKA	67
LAMPIRAN	69
RIWAYAT HIDUP	81

DAFTAR GAMBAR

Gambar 2.1	Kontruksi Motor DC Sederhana.....	5
Gambar 2.2	Ilustrasi Kerja RFID.....	6
Gambar 2.3	Mikrokontroler.....	9
Gambar 2.4	AT-MEGA 32.....	10
Gambar 2.5	Arsitektur Internal ATmega32.....	12
Gambar 2.6	Cara kerja sensor ultrasonic	13
Gambar 2.7	Sensor <i>Ultrasonic</i>	13
Gambar 2.8	Kontruksi Sensor Kecepatan.....	14
Gambar 2.9	Diagram Blok Kontroler PID.....	15
Gambar 2.10	Sinyal PWM	17
Gambar 2.11	Perubahan Lebar Pulsa Sinyal PWM	18
Gambar 2.12	Lebar Pulsa PWM berdasarkan Presentase <i>Duty Cycle</i>	19
Gambar 2.13	Skema Perhitungan Tegangan Output Rata-Rata	20
Gambar 3.1	Blok Diagram Fungsional Sistem	21
Gambar 3.2	Arsitektur Sistem	22
Gambar 3.3	Desain Pintu Geser.....	23
Gambar 3.4	Komponen dan Alur Perangkat Keras Sistem.....	25
Gambar 3.5	Motor DC yang telah dipasang gear dan roller chain	26
Gambar 3.6	Rangkaian Catu Daya Mikrokontroler.....	27
Gambar 3.7	Rangkaian Mikrokontroler AT-Mega32.....	28
Gambar 3.8	Rangkaian LCD	29
Gambar 3.9	Rangkaian H-Bridge L298N.....	30
Gambar 3.10	Optocoupler “U” S53	31
Gambar 3.11	Rangkaian Rotary Encoder Dengan Optocoupler “U” S53.....	32
Gambar 3.12	Kontruksi Sensor Kecepatan.....	32
Gambar 3.13	Ultrasonic HC-SRF04	33
Gambar 3.14	RFID ID-12	34
Gambar 3.15	Spesifikasi pin pada ID-2, ID-12, dan ID-20	35
Gambar 3.16	Rangkaian RFID Reader Menggunakan ID-12.....	35
Gambar 3.17	<i>Flowchart</i> Sistem Keseluruhan	36
Gambar 3.18	<i>Flowchart</i> Pembacaan ID pada RFID Tags	37
Gambar 3.19	<i>Flowchart</i> Pembacaan RPM	39
Gambar 3.20	<i>Flowchart</i> Pengendalian <i>Driver</i> Motor DC	40
Gambar 3.21	Grafik Respon Motor DC Tanpa Beban	41

Gambar 3.22	Grafik Respon Motor DC Berbeban.....	41
Gambar 3.23	Blok Diagram Kontrol PID.....	42
Gambar 3.24	Respon Tangga Satuan Sistem.....	43
Gambar 3.25	Kurva Respon Berbentuk S.....	43
Gambar 3.26	Kurva Respon Berbeban.....	44
Gambar 4.1	Integrasi Sistem Dengan <i>Prototype</i> Pintu Geser	47
Gambar 4.2	Integrasi Hardware	48
Gambar 4.3	Pengukuran Tegangan Pada Masing – Masing <i>Port</i>	50
Gambar 4.4	Mekanisme Pengujian Sinyal PWM Terhadap Output Driver Motor DC	51
Gambar 4.5	Pengukuran Sinyal PWM (20%, 40%, 60%, 80%)... ..	52
Gambar 4.6	Grafik Data Pengujian PWM Mikrokontroler Terhadap Tegangan Keluaran Driver Motor.....	53
Gambar 4.7	Mekanisme Pengujian Sensor Jarak	53
Gambar 4.8	Grafik Data Pengujian Perbandingan Sensor Jarak Dengan Alat Ukur Konvensional	55
Gambar 4.9	Mekanisme Pengujian Kecepatan Motor DC.....	56
Gambar 4.10	Mekanisme Pengujian RFID.....	57
Gambar 4.11	Pengujian Dengan Halangan <i>RFID</i>	59
Gambar 4.12	Kurva Perbandingan Kecepatan Motor DC Tanpa kontroler Dan Kontroler PID Terhadap Set Point	60
Gambar 4.13	Respon motor DC sebelum dikontrol terhadap set point.....	61
Gambar 4.14	Respon motor DC setelah dikontrol terhadap set point	61
Gambar 4.15	Respon Kecepatan Pintu Yang Berubah Saat Semakin Mendekati Batas Buka Pintu.....	63

DAFTAR TABEL

Tabel 3.1	Tabel Pin Konfigurasi dari LCD	29
Tabel 3.2	Tabel Kebenaran (Truth Table) L298N	31
Tabel 3.3	Konfigurasi Pin Sensor Ultrasonik	33
Tabel 3.4	Spesifikasi modul RFID reader ID-12	34
Tabel 4.1	Hasil Pengukuran Tiap <i>Port</i> Pada Sistem Minimum	50
Tabel 4.2	Data Pengujian PWM Mikrokontroler	52
Tabel 4.3	Data Pengujian Sensor Jarak	54
Tabel 4.4	Hasil Pengujian Kecepatan Motor DC	56
Tabel 4.5	Data Jarak Deteksi RFID	58
Tabel 4.6	Pengujian RFID Dengan Mengubah Posisi	58
Tabel 4.7	Pengujian RFID Dengan Penghalang	59
Tabel 4.8	Pengujian Integrasi Sistem	62

--Halaman ini sengaja dikosongkan--

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini telah banyak memberikan manfaat dan kemudahan bagi kehidupan manusia sehari-hari, khususnya perkembangan teknologi pada bidang elektronika. Adanya Pengaruh perkembangan teknologi pada bidang elektronika ini menimbulkan keinginan manusia untuk mengendalikan suatu peralatan elektronika secara otomatis. Hal tersebut dapat dilakukan melalui pengaturan level tegangan dan arus. Beberapa contoh pengontrolan secara otomatis yaitu pengontrolan putaran motor, pengaturan pencahayaan pada lampu, dan lain-lain. Mobilitas dan kesibukan yang dimiliki manusia juga mengakibatkan manusia ingin mendapatkan kemudahan dalam berbagai kegiatannya.

Salah satu contoh dari hasil pengamatan di lingkungan sekitar adalah pengendalian pintu garasi yang masih dilakukan secara manual. orang harus turun dari kendaraannya untuk membuka maupun menutup pintu garasi saat hendak memasukkan atau mengeluarkan mobilnya. *Remote control* sering kali digunakan untuk mempermudah dalam pengendalian pintu garasi tersebut.

Berdasarkan keadaan seperti itu, ada cara yang lebih efektif yang dapat dilakukan untuk mengatasi permasalahan tersebut. Salah satu cara yaitu pembangunan sebuah sistem dengan memanfaatkan kartu RFID. Kartu ini digunakan untuk mengenali identitas pengguna berdasarkan ID pada RFID Tags dan juga untuk mengendalikan pintu geser pada garasi rumah secara otomatis. Adanya konsep ini pemilik rumah dapat memudahkan, sehingga tidak perlu menggunakan *remote control* ataupun harus membuka pintu garasi secara manual.

Oleh karena itu, pada Tugas Akhir ini akan dirancang sebuah sistem menggunakan mikrokontroler dan RFID yang dapat mengenali identitas (ID) kendaraan dan untuk mengendalikan pintu geser garasi secara otomatis.

1.2 Perumusan Masalah

Pada Tugas Akhir ini, masalah yang dibahas mengenai beberapa kesulitan dalam perancangan sistem pengendali pintu geser otomatis

menggunakan mikrokontroler dan pembuatan sistem untuk mengenali ID pada RFID Tags sebagai identifikasi pengguna.

1.3 Batasan Masalah

Berdasarkan perumusan masalah di atas, beberapa hal yang perlu dibatasi dalam Tugas Akhir ini agar penelitian yang dilakukan dapat tercapai, yaitu:

1. Pada Sistem ini menggunakan RFID *reder* ID-12LA yang memiliki frekuensi baca 125 - 134 Khz atau *Low Frequency (LF)* dengan jarak baca max 12 cm.
2. Objek yang dikontrol merupakan miniatur dari pintu geser.
3. Perancangan sistem meliputi perancangan perangkat keras berbasis mikrokontroler.

1.4 Tujuan

Tujuan dari Tugas Akhir ini dibedakan menjadi tujuan umum dan tujuan khusus. Tujuan umum pada tugas akhir ini untuk memenuhi persyaratan akademis menyelesaikan studi pada Program S1 Teknik Elektro Bidang Studi Elektronika ITS. Tujuan khusus tugas akhir ini untuk merancang sistem kontrol pintu geser otomatis dengan identifikasi ID pada RFID Tags berbasis mikrokontroler.

1.5 Metodologi

Metodologi dalam pelaksanaan penelitian ini dilakukan dengan cara sebagai berikut:

1. Studi Literatur.
Kegiatan ini berupa mencari informasi baik dari literatur di perpustakaan, internet ataupun survey lapangan tentang pengendalian pintu geser, mikrokontroler, RFID, sensor jarak, serta kontroler yang akan digunakan.
2. Perancangan dan Pembuatan Sistem
Perancangan disusun atas dasar model sistem kontrol pintu geser otomatis dengan identifikasi ID pada kendaraan berbasis mikrokontroler. Perencanaan dan pembuatan aplikasi ini akan dilaksanakan di laboratorium elektronika.
3. Pengujian Sistem

Pengujian terhadap alat dan mengambil data hasil percobaan dilakukan di laboratorium elektronika..

4. Analisa Data

Berdasarkan hasil simulasi dan pengujian, dilakukan analisa terhadap persen kesalahan yang terjadi. Hal ini diperlukan untuk mengevaluasi kembali dari kontroler yang telah didesain.

5. Penulisan Buku Tugas Akhir

Buku tugas akhir ditulis secara intensif bila proses pengujian telah selesai.

1.6 Sistematika Penulisan

Sistematika penulisan yang disusun pada Tugas Akhir ini terdiri dari lima bab, sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, sistematika penulisan dan relevansi.

BAB II : DASAR TEORI

Bab ini berisi tentang dasar – dasar pengetahuan dari tiap komponen yang digunakan pada sistem yang akan dirancang seperti: teori motor DC, identifikasi sistem, *inverter*, sensor *rotary encoder*, dan PLC Mitsubishi.

BAB III : PERANCANGAN SISTEM

Bab ini berisi tentang perancangan sistem secara garis besar yang terdiri dari perancangan perangkat keras (*hardware*), perancangan perangkat lunak (*software*), identifikasi sistem, dan perancangan kontroler.

BAB IV : IMPLEMENTASI DAN ANALISA

Bab ini menjelaskan tentang hasil implementasi dan pengujian sistem secara keseluruhan.

BAB V : PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian Tugas Akhir ini dan saran – saran mengenai kemungkinan pengembangan penelitian ini.

1.7 Relevansi

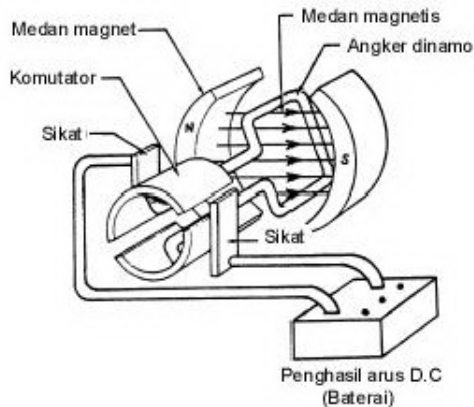
Hasil yang dicapai dari Tugas Akhir ini diharapkan menjadi salah

satu referensi yang dapat digunakan dalam memudahkan pekerjaan dan mobilitas manusia terutama dalam otomatisasi pintu geser.

BAB 2 DASAR TEORI

2.1 Motor Arus Searah (Motor DC)

Motor arus searah atau motor DC adalah suatu mesin yang berfungsi mengubah energi listrik arus searah menjadi energi mekanis[1]. Pada prinsipnya motor arus searah dapat digunakan sebagai generator arus searah, demikian pula sebaliknya generator arus searah dapat dipakai sebagai motor arus searah.



Gambar 2.1 Kontruksi Motor DC Sederhana[1]

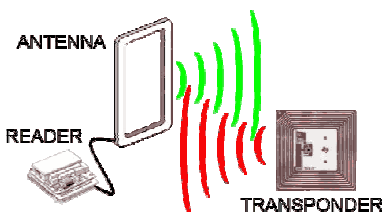
Bagian utama motor DC adalah stator dan rotor, dimana kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Bentuk kontruksi motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan seperti yang ditunjukkan pada Gambar 2.1 disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet. Catu tegangan dc dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung

lilitan. Kumparan satu lilitan pada gambar di atas disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

2.2 *Radio Frequency Identification (RFID)*

Radio Frequency Identification (RFID) adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut *tag* atau label RFID yang dipasang pada objek untuk diidentifikasi[2]. *Tag* atau label RFID adalah sebuah benda yang bisa dipasang atau dimasukkan di dalam sebuah produk, hewan atau bahkan manusia dengan tujuan untuk identifikasi menggunakan gelombang radio. *Tag* RFID berisi informasi yang disimpan secara elektronik dan dapat dibaca hingga beberapa meter jauhnya.

Informasi *Tag* disimpan secara elektronik di dalam memori *non-volatile*. *Tag* RFID mencakup pemancar dan penerima frekuensi radio kecil. Sebuah pembaca RFID mengirimkan sinyal radio yang dikodekan untuk memeriksa *tag*. Lalu, *tag* menerima pesan dan merespon informasi yang diidentifikasinya. Ilustrasi kerja secara sederhana dari RFID ditunjukkan pada Gambar 2.2.



Gambar 2.2 Ilustrasi Kerja RFID[2]

Perangkat RFID diatas akan berkomunikasi jika bekerja pada frekwensi yang sama. Sejalan dengan karakteristik frekwensi yang berbeda-beda dan kebutuhan atau kondisi lapangan yang juga sangat bervariasi dalam penerapan RFID. RFID menggunakan standar Global, yaitu EPC (Electronic Product Code). Standar ini mewajibkan setiap Tag yang dibuat harus memiliki Unique Identifier. Sehingga akan memudahkan dalam proses identifikasi objek tertentu secara spesifik. ID ini merupakan

kunci untuk merujuk pada data suatu objek yang ada pada database. Saat ini telah berkembang solusi RFID untuk 4 macam *frequency band* :

- *Low Frequency* (LF) : 125 - 134 KHz
Low frequency biasanya digunakan untuk identifikasi jarak dekat (dibaca dalam jarak hingga 30 cm) dan mampu menembus objek seperti dinding, tetapi tidak untuk metal. Beroperasi pada 125 kHz atau 134 kHz.
- *High Frequency* (HF) : 13.56 Mhz
High frequency yang memiliki jarak identifikasi yang lebih jauh (dibaca dalam jarak hingga 1 m) dan memiliki kecepatan yang lebih baik. Beroperasi pada frekwensi 13.56 MHz
- *Ultra High Frequency* (UHF) : 868 - 956 Mhz
Ultra high frequency untuk identifikasi jarak jauh dan lebih cepat. Namun proses identifikasinya tidak mampu menembus objek yang memiliki kandungan air tinggi. Beroperasi pada 866 MHz hingga 960MHz. UHF hanya mampu beroperasi pada jarak lebih dari 3,3 meter.
- *Microwave* : 2.45 Ghz

2.2.1 RFID Tag

RFID transponder atau RFID tag terdiri dari chip rangkaian sirkuit yang terintegrasi dan sebuah antena. Rangkaian elektronik dari RFID tag umumnya memiliki memori. Memori ini memungkinkan RFID tag mempunyai kemampuan untuk menyimpan data. Memori pada tag dibagi menjadi sel-sel. Beberapa sel menyimpan data *Read Only*, seperti ID number. Semua RFID tag mendapatkan ID number pada saat tag tersebut diproduksi.

Selain *Read Only* pada RFID tag memungkinkan RFID tag tersebut dapat ditulis (*Write*) dan dibaca secara berulang. Setiap tag dapat membawa informasi yang unik, seperti ID number, tanggal lahir, alamat, jabatan, dan data lain dari objek yang akan diidentifikasi. Banyaknya informasi yang dapat disimpan oleh RFID tag tergantung pada kapasitas memori nya. Semakin banyak fungsi yang dapat dilakukan oleh RFID tag maka rangkainnya akan semakin kompleks dan ukurannya akan semakin besar. Berdasarkan itu, RFID Tag digolongkan menjadi:

1. Tag Aktif

Tag ini dapat dibaca (*Read*) dan ditulis (*Write*). Baterai yang terdapat di dalam tag ini digunakan untuk memancarkan gelombang radio kepada reader sehingga reader dapat membaca data yang

terdapat pada tag ini. Dengan adanya internal baterai, tag ini dapat mengirimkan informasi dalam jarak yang lebih jauh dan reader hanya membutuhkan daya yang kecil untuk membaca tag ini. Kelemahan dari tipe tag ini adalah harganya yang mahal dan ukurannya yang lebih besar.

2. Tag Pasif

Tag ini hanya dapat dibaca saja (Read) dan tidak memiliki internal baterai seperti halnya tag aktif. Sumber tenaga untuk mengaktifkan tag ini didapat dari RFID reader. Ketika medan gelombang radio dari reader didekati oleh tag pasif, koil antena yang terdapat pada tag pasif ini akan membentuk suatu medan magnet. Medan magnet ini akan menginduksi suatu tegangan listrik yang memberi tenaga pada tag pasif. Keuntungan dari tag ini adalah rangkaiannya lebih sederhana, harganya jauh lebih murah, ukurannya lebih kecil, dan lebih ringan. Kelemahannya adalah tag hanya dapat mengirimkan informasi dalam jarak yang dekat dan untuk membaca tag ini, RFID reader harus memancarkan gelombang radio yang cukup besar sehingga menggunakan daya yang cukup besar.

2.2.2 RFID Reader

RFID reader adalah merupakan penghubung antara software aplikasi dengan antena yang akan meradiasikan gelombang radio ke RFID tag. Gelombang radio yang ditransmisikan oleh antena berpropagasi pada ruangan di sekitarnya. Akibatnya data dapat berpindah secara wireless ke tag RFID yang berada berdekatan dengan antena.

2.3. Mikrokontroler

Mikrokontroler adalah sebuah IC yang di dalamnya terdapat mikroprosesor dan memori program (ROM) serta memori serbaguna (RAM)[3]. Bahkan ada jenis mikrokontroler yang memiliki fasilitas ADC, PLL, EEPROM, dalam satu kemasan. Penggunaan mikrokontroler dalam bidang Kontrol sangat luas dan populer. Ada beberapa vendor yang membuat mikrokontroler diantaranya *Intel*, *Microchip*, *Windbond*, *Atmel*, *Philips*, *Xemics* dan lain-lain.

AVR adalah chip mikrokontroler yang merupakan modifikasi dari arsitektur harvard 8-bit RISC. AVR adalah salah satu dari keluarga mikrokontroler yang menggunakan flash memory on-chip untuk penyimpanan program. Komponen mikrokontroler ditunjukkan pada

Gambar 2.3.



Gambar 2.3 Mikrokontroler

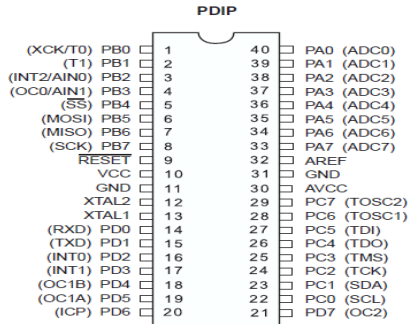
AVR memiliki 32 register *general-purpose*, *timer/counter* fleksibel dengan mode *compare*, interrupt internal dan eksternal, serial UART, programmable watchdog timer, dan mode *power saving*, bahkan beberapa AVR sudah ada yang memiliki ADC dan PWM internal. Selain itu, AVR juga memiliki *In-System Programmable Flash On-Chip* yang memungkinkan mikrokontroler ini diprogram ketika mikrokontroler sudah terpasang pada *board* dalam sistem. *On-chip programming* ini menggunakan serial SPI dalam realisasinya.

Secara umum AVR dibagi menjadi 3 kelompok besar yaitu :

- a. Keluarga AVR Tiny
 - 1 – 8 Kbyte memori program
 - 8 – 20 kemasan jumlah pin
 - Peripheral yang terbatas
 - b. Keluarga AVR Mega
 - 4 – 256 Kbyte memori program
 - 28 – 100 kemasan pin
 - *Peripheral* lebih banyak
 - c. Keluarga AVR untuk aplikasi yang spesifik
- Keluarga AVR mega dengan fitur spesial yang tidak ditemukan pada keluarga AVR di atas seperti, kontroler LCD, Kontroler USB, advance PWM, dan lain-lain. Untuk memprogram Flash-ROM dari

mikrokontroler keluarga Atmel AVR ini digunakan bahasa C dengan *compiler Code Vision AVR versi 1.25.0*.

2.3.1 Mikrokontroler AT-Mega32

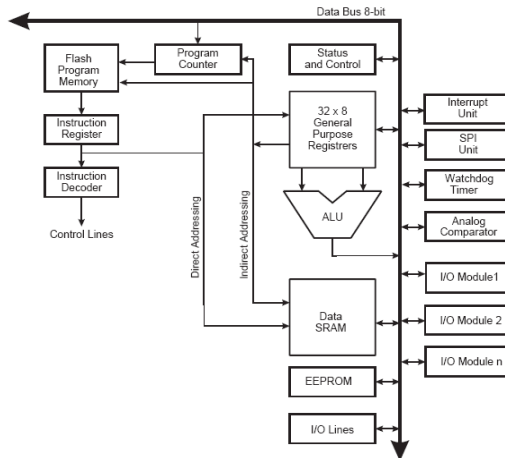


Gambar 2.4 AT-MEGA 32[4]

Deskripsi dari pin-pin AT-Mega32 adalah sebagai berikut :

1. **VCC:** Tegangan *supply*
2. **GND:** Ground
3. **Port B (PB7..PB0):** Port I/O 8 –bit dengan resistor *Pull Up* internal tiap pin, *Buffer Port B* mempunyai kapasitas menyerap (*sink*) dan Mencatu (*source*).
4. **Port C (PC7..PC0):** Port I/O 8-bit ([PC6], PC5...PC0) dengan resistor *Pull-Up* internal tiap pin. *Buffer Port C* mempunyai kapasitas menyerap (*sink*) dan mencatu (*source*).
5. **Port D (PD7..PD0):** Port I/O 8-bit dengan resistor *Pull-Up* internal tiap pin. *Buffer Port C* mempunyai kapasitas menyerap (*sink*) dan mencatu (*source*).
6. **AVcc:** AVcc adalah pin tegangan catu untuk A/D *converter*. AVcc harus dihubungkan ke Vcc, walaupun ADC tidak digunakan. Jika ADC digunakan , maka AVcc harus dihubungkan ke VCC melalui “*low pass filter*”.
7. **AREF:** untuk pin tegangan referensi analog untuk ADC.
8. **ADC7.ADC0(TQPF,QFN/MLF):** Digunakan untuk *input* konverter analog to digital.

9. **Reset:** Sebuah *low level* pulsa yang lebih lama daripada lebar pulsa minimum pada pin ini akan menghasilkan *reset* meskipun *clock* tidak berjalan.
10. **XTAL1:** *Input inverting* penguat *Oscillator* dan *input* internal clock operasi rangkaian.
11. **XTAL2:** *Output* dari *inverting* penguat *Oscillator*.



Gambar 2.5 Arsitektur Internal ATmega32[4]

Spesifikasi internal dari mikrokontroler ATmega32 adalah sebagai berikut:

- 8-bit *Central Processing Unit* (CPU).
- 32 x 8 *General Purpose Working Register*.
- 16 MIPS (*Million Instruction Per Second*) per 16 MHz, untuk satu instruksi memerlukan satu siklus waktu.
- 32KBytes of *In-System Reprogrammable Flash Program Memory* yang dapat diisi hingga 1000kali.
- 1024 bytes EEPROM yang dapat diisi hingga 100.000 kali.
- 2 KBytes *internal SRAM*.
- 32 jalur masukan dan data keluaran yang bersifat dua arah yang dikelompokkan menjadi 4 port 8-bit (PORTA...D).
- Siklus waktu operasi hingga 16MHz.

- *Full duplex serial port*,
- 2 buah pewaktu 8-bit,
- 1 buah pewaktu 16-bit,
- *Real Time Counter* (RTC) dengan osilator yang terpisah,
- 8 kanal 10-bit ADC,
- *Master/Slave* antar muka SPI serial,
- *Analog comparator*,
- Membutuhkan tegangan catu sebesar 2.7 – 5.5 V untuk tipe ATmega 32L dan 4.5 – 5.5 V untuk ATmega 32.

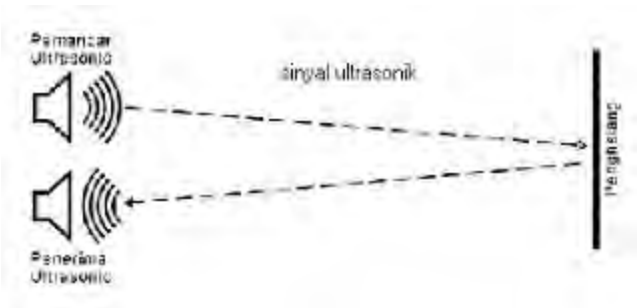
2.4 Sensor Jarak

Sensor jarak adalah sebuah sensor mampu mendeteksi keberadaan benda di dekatnya tanpa kontak fisik[5]. Sensor jarak sering memancarkan elektromagnetik atau berkas radiasi elektromagnetik (inframerah, misalnya), dan mencari perubahan dalam bidang atau sinyal kembali. Objek yang sedang merasakan sering disebut sebagai sensor jarak target itu.

2.4.1 Sensor Ultrasonik

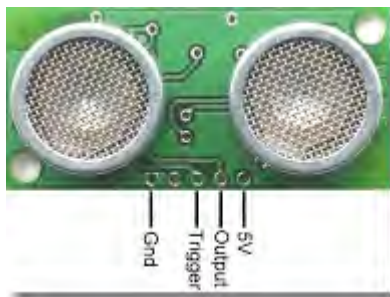
Sensor ultrasonik bekerja berdasarkan prinsip pantulan gelombang suara, dimana sensor ini menghasilkan gelombang suara yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar pengindraannya. Perbedaan waktu antara gelombang suara dipancarkan dengan ditangkapnya kembali gelombang suara tersebut adalah berbanding lurus dengan jarak atau tinggi objek yang memantulkannya. Jenis objek yang dapat diindra diantaranya adalah: objek padat, cair, butiran maupun tekstil.

Sensor *ultrasonic* adalah sebuah sensor yang memanfaatkan pancaran gelombang *ultrasonic*. Sensor *ultrasonic* ini terdiri dari rangkaian pemancar *ultrasonic* yang disebut *transmitter* dan rangkaian penerima *ultrasonic* disebut *receiver*.



Gambar 2.6 Cara kerja sensor ultrasonic

Ultrasonic modul umumnya berbentuk papan elektronik ukuran kecil dengan beberapa rangkaian elektronik dan 2 buah transducer. Dari 2 buah transducer ini, salah satu berfungsi sebagai transmitter dan satu lagi sebagai receiver. Ada juga modul yang hanya mempunyai 1 buah transducer, berfungsi sebagai transmitter dan receiver sekaligus. Tersedia pin VCC, TRIG, ECHO dan GND. Ada juga modul yang pin TRIG dan ECHO-nya digabung menjadi satu dan penggunaannya berganti-ganti.



Gambar 2.7 Sensor Ultrasonic[6].

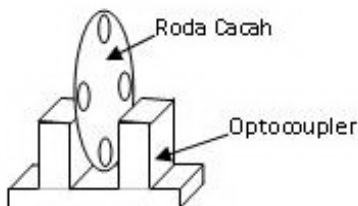
Sensor ini dapat mengukur jarak antara 2 cm sampai 400 cm. keluaran dari sensor ini berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 μ S sampai 18,5 mS. Sensor *ultrasonic ping parallax* terdiri dari sebuah *chip* pembangkit sinyal 40KHz, sebuah *speaker ultrasonic* dan sebuah *microphone ultrasonic*. *Speaker*

ultrasonic mengubah sinyal 40 KHz menjadi suara sementara *microphone ultrasonic* berfungsi untuk mendeteksi pantulan suaranya

2.5 Sensor Kecepatan

Sensor kecepatan atau velocity sensor merupakan suatu sensor yang digunakan untuk mendeteksi kecepatan gerak benda untuk selanjutnya diubah kedalam bentuk sinyal elektrik[5]. Dalam prakteknya ada beberapa sensor yang digunakan untuk berbagai keperluan ini, salah satunya dengan menggunakan optocoupler tipe “U” dan sebuah roda cacah. Sensor putaran atau kecepatan ini dapat digunakan untuk membaca putaran suatu object yang berputar seperti roda kendaraan, putaran motor listrik dan lain nya. Sensor putaran atau kecepatan ini dibuat dengan optocoupler tipe “U” yang ditengahnya diletakan sebuah roda cacah. Optocoupler merupakan komponen optoisolator yang memiliki karakteristik penerima (photo transistor) akan mengalami perubahan logika bila terjadi perubahan intensitas cahaya yang dipancarkan oleh pemancar (LED infra merah) untuk penerima.

Roda cacah yang diletakan ditengah optocoupler tersebut berfungsi untuk mempengaruhi intensitas cahaya yang diberikan oleh LED pada optocoupler ke photo transistor yang akan memberikan perubahan level logika sesuai dengan putaran roda cacah. Kecepatan perubahan logika photo transistor akan sebanding dengan kecepatan putaran roda cacah. Konstruksi sensor putaran dapat dilihat pada gambar berikut. Konstruksi Sensor Putaran Rotary encoder, atau disebut juga Shaft encoder, merupakan perangkat elektromekanikal yang digunakan untuk mengkonversi posisi anguler (sudut) dari shaft (lubang) atau roda ke dalam kode digital, menjadikannya semacam transduser.



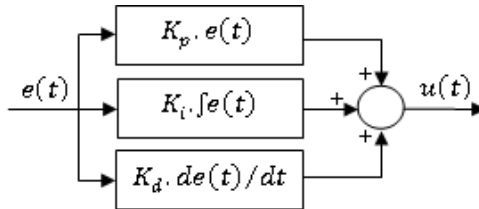
Gambar 2.8 Kontruksi Sensor Kecepatan

2.6 Kontroler PID

Kontroler memiliki peranan yang besar dalam suatu sistem kontrol. Adanya kontroler dapat mengatur dan memperbaiki sistem kerja yang terdapat pada suatu sistem kontrol. Prinsip kerja dari kontroler adalah memperkecil sinyal kesalahan yang terdapat pada sistem (mengurangi perbedaan antara sinyal *setting* dan sinyal aktual) sehingga sinyal keluaran dapat diatur sesuai dengan nilai sinyal *setting* yang dikehendaki.

Salah satu kontroler yang digunakan adalah kontroler PID. Kontrol PID merupakan algoritma metode kendali yang terdiri dari tiga elemen yaitu proporsional (P), integral (I), dan *derivative* (D). Diagram blok dari kontroler PID dapat dilihat pada Gambar 2.5. Kontroler PID seperti yang ditunjukkan pada Persamaan 2.1.

$$u(t) = K_p(e(t) + \frac{1}{\tau_i} \int_0^t e(t) + \tau_d \frac{de(t)}{dt}) \quad (2.1)$$



Gambar 2.9 Diagram Blok Kontroler PID[7]

2.5.1 Kontroler Proporsional (P)

Kontroler proporsional merupakan kontroler yang aksi kontrolnya proporsional atau sebanding terhadap sinyal kesalahan^[8]. Kontroler proporsional berupa konstanta dan secara sederhana keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Persamaan dari kontrol proporsional dapat dilihat sebagai berikut:

$$u(t) = K_p(e(t)) \quad (2.2)$$

Karakteristik dari kontroler proporsional adalah dapat mempercepat respon sistem untuk mencapai keadaan *steady state*. Semakin besar nilai

kontrol proporsional akan memperkecil kesalahan keadaan *steady state*, namun akan membawa sistem ke daerah tidak stabil, karena itu kontroler proporsional tidak dapat menghilangkan nilai *error* ($E_{ss} \neq 0$), sehingga diperlukan kontroler integral agar nilai *error* dapat mendekati nol.

2.5.2 Kontroler Integral (I)

Karakteristik kontroler integral adalah menghasilkan respon sistem yang memiliki nilai *error steady state* = 0 (*error* mendekati nol), namun kontroler ini cenderung memperlambat respon^[8]. Keluaran kontroler integral merupakan penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Persamaan dari kontrol integral dapat dilihat sebagai berikut:

$$u(t) = \frac{K_p}{\tau_i} \int_0^t e(t) dt \quad (2.3)$$

2.5.3 Kontroler Derivatif (D)

Karakteristik keluaran kontroler diferensial memiliki sifat seperti operasi derivatif^[8]. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat pada keluaran. Persamaan dari kontrol diferensial dapat dilihat sebagai berikut:

$$u(t) = K_p \tau_d \left(\frac{de(t)}{dt} \right) \quad (2.4)$$

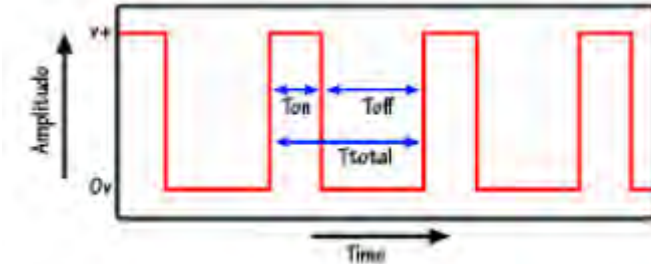
Kelebihan kontroler derivatif apabila ditambah dengan kontroler proporsional dapat mengurangi aksi kontrol yang berlebihan sehingga tidak mengalami *overshoot*.

2.6 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu perioda, untuk mendapatkan tegangan rata-rata yang berbeda. Modulasi PWM dilakukan dengan cara merubah lebar pulsa dari suatu pulsa data^[9]. Total 1 perioda (T) pulsa dalam PWM adalah tetap, dan data PWM pada umumnya menggunakan perbandingan pulsa positif terhadap total pulsa.

Pada metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Misalkan PWM digital 8 bit berarti PWM tersebut

memiliki resolusi $2^8 = 256$, maksudnya nilai keluaran PWM ini memiliki 256 variasi, variasinya mulai dari 0 – 255 yang mewakili *duty cycle* 0 – 100% dari keluaran PWM tersebut.



Gambar 2.10 Sinyal PWM[9]

Sinyal PWM pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar Pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Artinya, Sinyal PWM memiliki frekuensi gelombang yang tetap namun *duty cycle* bervariasi (antara 0% hingga 100%)

$$D = \frac{T_{on}}{T_{total}}; \text{ dimana } T_{total} = T_{on} + T_{off} \quad (2.5)$$

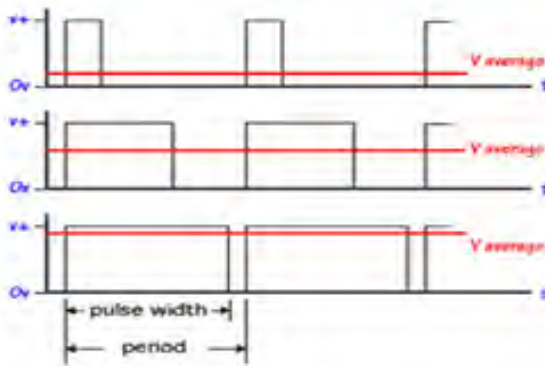
$$V_{out} = D \times V_{in} \quad (2.6)$$

$$V_{out} = \frac{T_{on}}{T_{off}} \times V_{in} \quad (2.7)$$

Keterangan :

- T_{on} = Waktu Pulsa “High”
- T_{off} = Waktu Pulsa “Low”
- $D = \text{Duty Cycle}$ adalah lamanya pulsa “High” dalam satu periode

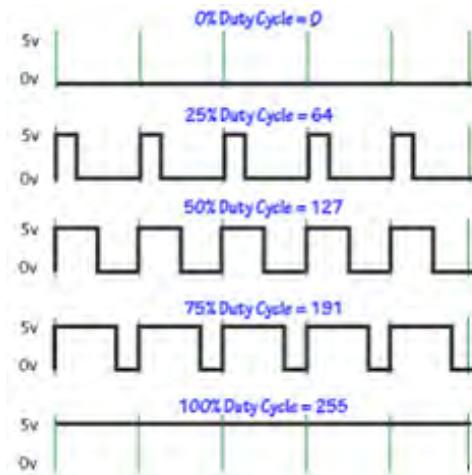
Dari persamaan diatas diketahui bahwa perubahan *duty cycle* akan merubah tegangan keluaran atau tegangan rata-rata seperti gambar dibawah ini



Gambar 2.11 Perubahan Lebar Pulsa Sinyal PWM[9]

Pulse Width Modulation (PWM) merupakan salah satu teknik untuk mendapatkan signal analog dari sebuah piranti digital. Sebenarnya Sinyal PWM dapat dibangkitkan dengan banyak cara, dapat menggunakan metode analog dengan menggunakan rangkaian op-amp atau dengan menggunakan metode digital.

Dengan metode analog setiap perubahan PWM-nya sangat halus, sedangkan menggunakan metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Resolusi adalah jumlah variasi perubahan nilai dalam PWM tersebut. Misalkan suatu PWM memiliki resolusi 8 bit berarti PWM ini memiliki variasi perubahan nilai sebanyak $2^8 = 256$ variasi mulai dari 0 – 255 perubahan nilai yang mewakili duty cycle 0 – 100% dari keluaran PWM tersebut.



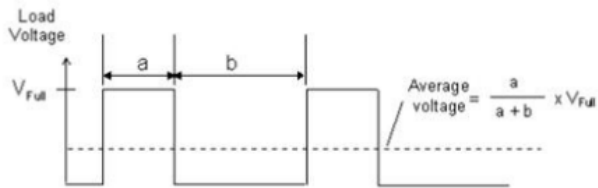
Gambar 2.12 Lebar Pulsa PWM berdasarkan Presentase *Duty Cycle*[9]

Dengan cara mengatur lebar pulsa “on” dan “off” dalam satu perioda gelombang melalui pemberian besar sinyal referensi output dari suatu PWM akan didapat *duty cycle* yang diinginkan. *Duty cycle* dari PWM dapat dinyatakan sebagai:

$$\text{DutyCycle} = \frac{\text{ton}}{(\text{ton} + \text{toff})} \times 100\% \quad (2.8)$$

Duty cycle 100% berarti sinyal tegangan pengatur motor dilewatkan seluruhnya. Jika tegangan catu 100V, maka motor akan mendapat tegangan 100V. pada *duty cycle* 50%, tegangan pada motor hanya akan diberikan 50% dari total tegangan yang ada, begitu seterusnya.

Perhitungan Pengontrolan tegangan output motor dengan metode PWM cukup sederhana. Dengan menghitung *Duty cycle* yang diberikan, akan didapat tegangan output yang dihasilkan. Sesuai dengan rumus yang telah dijelaskan pada gambar 2.13



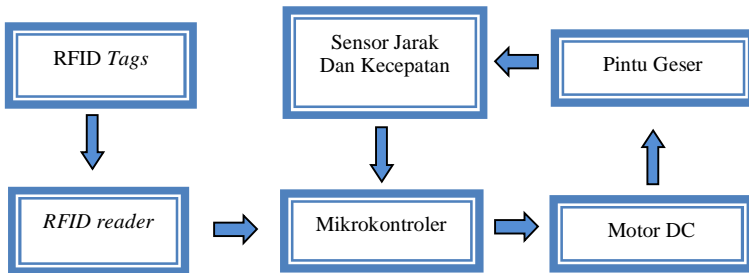
Gambar 2.13 Skema Perhitungan Tegangan Output Rata-Rata[9]

Average voltage merupakan tegangan output pada motor yang dikontrol oleh sinyal PWM. **a** adalah nilai *duty cycle* saat kondisi sinyal “on”. **b** adalah nilai *duty cycle* saat kondisi sinyal “off”. **V full** adalah tegangan maximum pada motor. Dengan menggunakan rumus diatas, maka akan didapatkan tegangan output sesuai dengan sinyal kontrol PWM yang dibangkitkan.

BAB 3

PERANCANGAN SISTEM

Pada perencanaan dan pembuatan perangkat proyek akhir ini mengacu pada diagram alur yang ditunjukkan pada Gambar 3.1 di bawah ini.



Gambar 3.1 Diagram Alur Fungsional Sistem

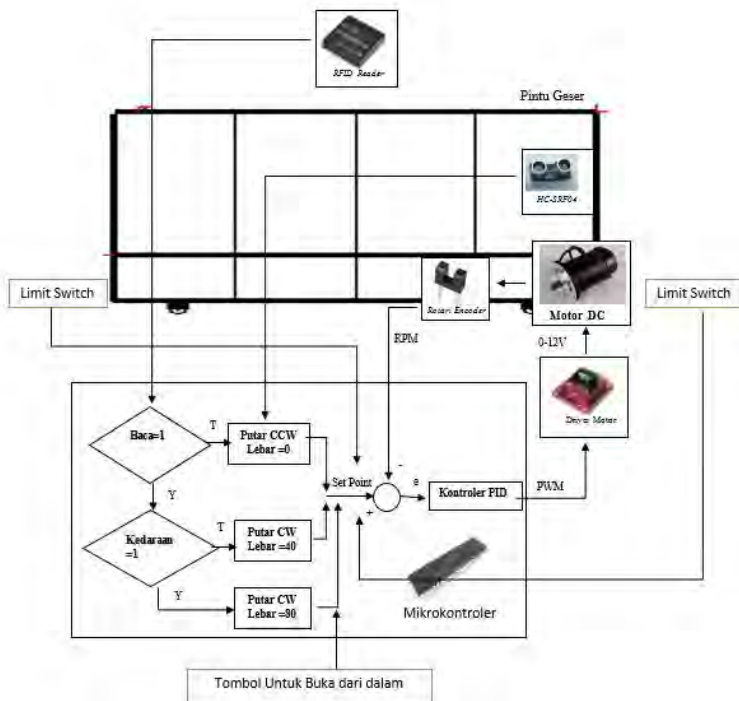
Blok diagram di atas dapat dilihat bahwa pada proyek akhir ini bermaksud untuk mengendalikan pintu geser secara otomatis dengan mengidentifikasi ID pengguna menggunakan RFID. Mikrokontroler digunakan sebagai kontroler dan pengolah data. Mikrokontroler akan mengendalikan pintu geser menggunakan motor DC sebagai penggerak pintu geser.

3.1 Arsitektur Sistem

Sistem yang dirancang adalah sistem mengenali ID pengguna dan mengendalikan pergerakan pintu geser. Sistem yang dirancang ini mempunyai tujuan agar dapat mengenali ID pengguna sehingga dapat diidentifikasi termasuk pengguna kendaraan atau manusia. Oleh karena itu, pintu geser dapat dikendalikan sesuai objek yang dideteksi. Objek yang diidentifikasi dikelompokkan dalam 2 golongan yakni objek manusia dan kendaraan beroda 4 (mobil). Prinsip kerja dari sistem motor DC ini akan mengendalikan lebar pintu geser yang akan dibuka berdasarkan objek yang dideteksi. Motor DC akan digerakan dengan kecepatan tertentu sesuai dengan *setpoint* yang telah ditentukan pada awalnya. Kecepatan dari motor DC akan dikurangi dengan

menggunakan pengendalian lebar pulsa input pada *driver* motor DC sehingga tegangan picu pada mototr DC akan berkurang dan kecepatan motor menurun.

Sistem yang dirancang terdiri dari beberapa komponen yang terhubung satu dengan lain. Komponen dan arsitektur dari sistem ditunjukkan seperti pada Gambar 3.2.



Gambar 3.2 Arsitektur Sistem

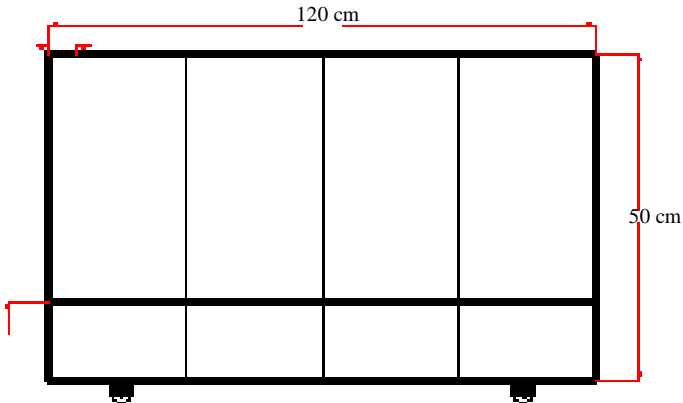
Komponen yang digunakan pada sistem ini, diantaranya RFID, mikrokontroler ATMEGA-32, *driver motor* L298N, sensor jarak HC-SRF04 untuk mendeteksi objek yang melalui pintu geser, motor DC, *rotary encoder* (Sharp S53), dan *limit switch*. Pada sistem ini, RFID digunakan sebagai identifikasi objek untuk menentukan lebar buka pintu

geser. Mikrokontroler sebagai perangkat yang digunakan untuk memberikan mekanisme kontrol pada motor DC sesuai dengan program yang diberikan. *Driver motor* L298N sebagai aktuator yang digunakan untuk pemberian aksi dalam pengaturan kecepatan motor DC dan arah putar motor DC. *Rotary encoder* sebagai sensor yang memberikan sinyal umpan balik pada sistem sehingga dapat mengetahui nilai *error* yang terjadi dalam sistem kontrol.

Perancangan sistem dibagi menjadi empat tahap. Tahap pertama ialah perancangan *prototype* pintu geser, perancangan perangkat keras (*hardware*) yang meliputi perancangan mekanik *plant*, rangkaian mikrokontroler TA-Mega32, *driver motor* DC, dan rangkaian *rotary encoder* menggunakan Sharp S53. Tahap kedua berupa *pemrograman* perangkat lunak (*software*). Tahap ketiga berupa identifikasi *plant* dan perancangan kontroler. Tahap yang terakhir yaitu integrasi sistem secara keseluruhan.

3.2 Perancangan *Prototype* Pintu Geser

Pintu garasi ini digunakan sebagai beban dari motor DC. Perencanaan ini dilakukan karena selain sebagai *plant*, juga untuk mengetahui spesifikasi dari motor DC yang akan digunakan sebagai penggerak. Pintu garasi yang digunakan ialah pintu dengan jenis geser. Desain pintu yang digunakan seperti yang ditunjukkan pada Gambar 3.3.



Gambar 3.3 Desain Pintu Geser

Untuk dimensi dari desain pintu yang direncanakan adalah sebagai berikut:

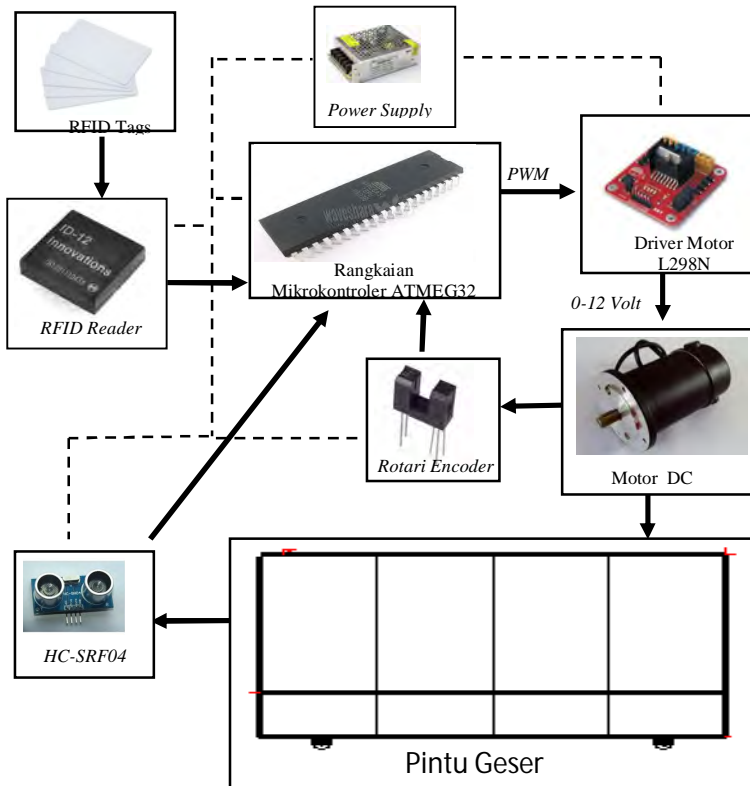
panjang	: 120 cm
tinggi	: 55 cm
jarak antar bilah di tengah	: 30 cm

Pada bilah horizontal bagian bawah ditempatkan *roler chain* (rantai) untuk dihubungkan dengan gear pada poros motor agar pintu geser dapat bergerak. Panjang dari rantai sama dengan panjang pintu geser sebesar 120 cm.

3.3 Perancangan Perangkat Keras (*Hardware*)

Perangkat keras (*Hardware*) dari sistem terdiri dari beberapa komponen. Komponen tersebut dirancang dan disesuaikan berdasarkan kebutuhan yang diperlukan pada *plant*. Komponen yang diperlukan pada *plant* diantaranya RFID, mikrokontroler AT-Mega32, *driver motor DC*, motor DC, *power supply*, sensor jarak HC-SRF04, dan *rotary encoder*. Perangkat keras mikrokontroler yang digunakan pada sistem adalah ATMEGA-32 dan *driver motor DC* yang digunakan adalah L298N.

Tahap pertama dalam perancangan sistem adalah perancangan perangkat keras (*hardware*) yang meliputi perancangan mekanik *plant* (motor DC penggerak), perancangan sistem minimum mikrokontroler, perancangan rangkaian driver motor, perancangan rangkaian *rotary encoder*, rangkaian RFID Reader, dan perancangan *wiring* pada *plant*. Sistem terdiri dari beberapa perangkat keras. Komponen dan alur dari perangkat keras sistem secara spesifik ditunjukkan pada Gambar 3.4.



Gambar 3.4 Komponen dan Alur Perangkat Keras Sistem

ID pengguna pada RFID Tags akan dideteksi oleh RFID reader. ID tersebut akan diidentifikasi oleh mikrokontroler sehingga apabila dikenali maka akan ditentukan golongan sebagai nilai set untuk lebar buka pintu geser. Selanjutnya mikrokontroler akan memberikan nilai set

driver motor untuk memutar searah dengan arah buka pintu. Selain itu mikrokontroler akan memberikan nilai set PWM 0-100% agar *driver* motor dapat menghasilkan tegangan yang berbeda sehingga kecepatan motor dapat dikendalikan.

Sensor jarak HHC-SRF04 digunakan untuk mengetahui ada tidaknya objek yang berada ditengan jalan masuk pintu geser. Sensor ini bekerja menggunakan gelombang ultrasonic. Nilai set kecepatan diberikan berdasarkan jarak lebar geser pintu dengan menggunakan perubahan jumlah pulsa pada *rotary encoder*. Sensor *rotary encoder* digunakan untuk mengukur kecepatan piringan yang berputar untuk selanjutnya diubah kedalam bentuk sinyal elektrik.. Keluaran dari sensor ini berupa pulsa dan hasil keluaran akan disalurkan ke pin *interrupt* (INT0) pada mikrokontroler. Hasil pembacaan kecepatan dari *rotary encoder* digunakan untuk mengetahui nilai *error* yang terjadi dan nantinya hasil tersebut digunakan untuk proses *kontrol* yang digunakan.

3.3.1 Perancangan Mekanik *Plant*

Plant yang digunakan dalam Tugas Akhir ini adalah motor DC 19 Volt. SPROCKET 40 1B digunakan sebagai gear untuk disabungkan ke *Roller Chain* yang disatukan dengan pagar agar dapat digerakan oleh motor DC. Berikut Tampilan motor DC dengan SPROCKET 40 1B yang meng-*coupe*l pintu geser ditunjukkan pada Gambar 3.5.

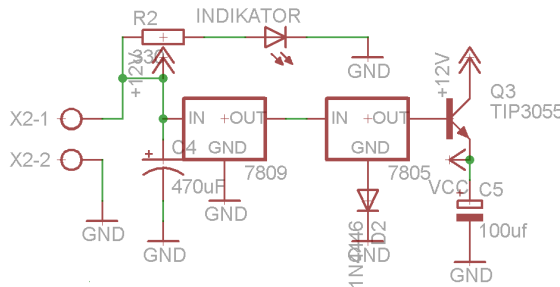


Gambar 3.5 Motor DC yang telah dipasang *gear* dan *roller chain*

3.3.2 Perancangan Rangkaian Mikrokontroler AT-Mega32

Rangkaian rangkaian mikrokontroler terdiri dari mikrokontroler AVR Atmel ATmega 32 dan berserta komponen-komponen pendukung

dan juga LCD 20x4 Karakter. Mikrokontroler AVR Atmel ATmega 32 merupakan otak penggerak dari keseluruhan sistem. Mikrokontroler memerlukan catu daya 5 Volt, sedangkan catu daya yang dimiliki sebesar 12 Volt sehingga diperlukan rangkaian untuk merubah catu daya dari 12 Volt ke 5 Volt. Berikut rangkaian catu daya mikrokontroler yang digunakan pada Gambar 3.6.



Gambar 3.6 Rangkaian Catu Daya Mikrokontroler

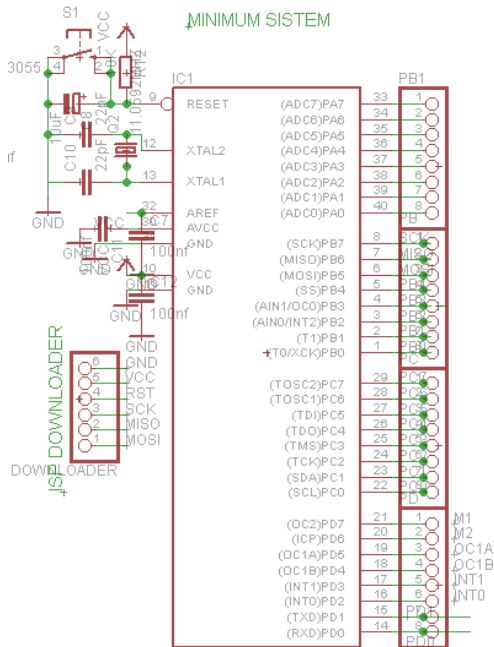
Rangkaian di atas mendapatkan tegangan masukan 12 Volt dan juga diberikan regulator tegangan sehingga tegangan keluaran menjadi menurun. Penggunaan 2 regulator tegangan yakni 7809 dan 7805 ditujukan agar tegangan yang dihilangkan pada regulator tidak terlalu besar sehingga menyebabkan IC Regulator panas.

Penggunaan masing – masing port I/O mikrokontroler ATmega32 dalam sistem ini adalah sebagai berikut :

1. PORT C: sebagai pembaca sensor jarak pada ultrasonic HC-SRF04. PIN C.0 sebagai *output trigger* dan PIN C.1 *input echo*.
2. PORT B: dihubungkan dengan *display LCD*
 - a. PIN B.0 terhubung ke PIN RS (*Register Select*)
 - b. PIN B.1 terhubung ke PIN R/W(*Read/Write*)
 - c. PIN B.2 terhubung ke PIN EN (*Enable LCD*)
 - d. PIN B.4 terhubung ke PIN D4 (*Data Bus*)
 - e. PIN B.5 terhubung ke PIN D5 (*Data Bus*)
 - f. PIN B.6 terhubung ke PIN D6 (*Data Bus*)
 - g. PIN B.7 terhubung ke PIN D7 (*Data Bus*)
3. PIN D.0 dan PIN D.1: sebagai *Transmit (TX)* dan *Receiver (RX)* untuk menerima data serial TTL dari RFID Reader.

4. PIN D.2: Sebagai *interrupt external* (INT0) untuk menerima input trigger dari sensor *rotary encoder*.
5. PIN D.5: sebagai *port output* PWM yang terhubung dengan PIN *enable* pada *driver* motor L298N
6. PIN D.6 dan D.7: sebagai output untuk pengendali arah putaran motor

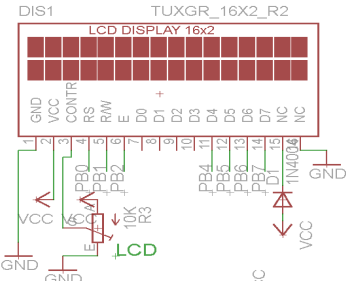
Rangkaian rangkaian mikrokontroler AT-Mega32 dari Tugas Akhir ini ditunjukkan pada Gambar 3.7.



Gambar 3.7 Rangkaian Mikrokontroler AT-Mega32

Liquid Crystal Display (LCD) atau dapat diterjemahkan dalam bahasa Indonesia sebagai tampilan Kristal Cair adalah suatu jenis media tampilan yang menggunakan Kristal Cair sebagai penampil utama. LCD bisa memunculkan gambar atau tulisan dikarenakan terdapat banyak

sekali titik cahaya (piksel) yang terdiri dari satu buah Kristal cair sebagai sebuah titik cahaya. Skematik LCD yang telah dibuat ditunjukkan pada Gambar 3.8.



Gambar 3.8 Rangkaian LCD

Konfigurasi dari fungsi kaki pada LCD *display* dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel Pin Konfigurasi dari LCD

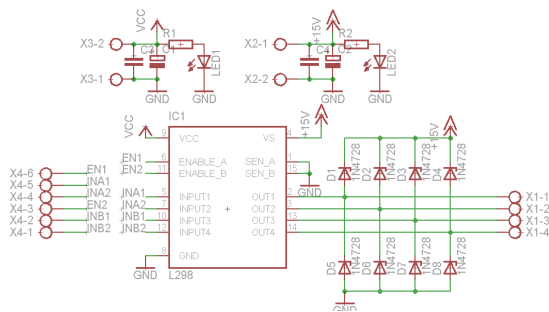
No.	Nama Pin	Deskripsi
1	GND	0V
2	VCC	+5V
3	VEE	Kontras LCD
4	RS	Register Select
5	R/W	1 = Read ; 0 = Write
6	EN	Enable LCD, 1=enable
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7
15	Anoda	Anoda Backlight LED
16	Katoda	Katoda Backlight LED

3.3.3 Perancangan Rangkaian *Driver Motor DC*

Rangkaian *driver motor DC* ini menggunakan IC L298N. IC *H Bridge driver motor DC* L298 memiliki dua buah rangkaian *H-Bridge* di dalamnya, sehingga dapat digunakan untuk men-*drive* dua buah motor DC. *H-Bridge driver motor DC* L298 masing-masing dapat mengantarkan arus hingga 2A. Pada Tugas Akhir ini, *H Bridge driver motor DC* L298 digunakan secara paralel, sehingga kemampuan menghantarkan dari *H-Bridge driver motor DC* L298 arusnya menjadi 4A. Spesifikasi IC *H-Bridge* L298N adalah sebagai berikut:

1. Tegangan operasi mencapai 46 volts [10].
 2. Mampu mengendalikan motor stepper bipolar 2 amps [10].
 3. Mampu mengontrol arah motor DC dengan arus maks 2 amps setiap h-bridge [10].
 4. Mampu mengendalikan motor DC 4 amp dengan memparalelkan kedua h-bridge di dalam IC L298 [10].
 5. Mendukung control PWM dengan frekuensi mencapai 20 KHZ
 6. Mempunyai sensor arus keluaran [10].
 7. Mempunyai Proteksi over temperature
- Supply voltage: +5 VDC [10].

Rangkaian dari *H-Bridge* L298N ditunjukkan pada Gambar 3.9.



Gambar 3.9 Rangkaian *H-Bridge* L298N

Pengoperasian IC *H-Bridge* L298N dapat dilihat pada tabel kebenaran yang ditunjukkan pada tabel 3.2 berikut:

Tabel 3.2 Tabel Kebenaran (Truth Table) L298N[10]

Input		Function
Venable=H	IN1=H ; IN2=L	Forward
	IN1=L ; IN2=H	Reverse
	IN1=IN2	Fast Motor Stop
Venable=L	IN1=x ; IN2=x	Free Running Motor Stop

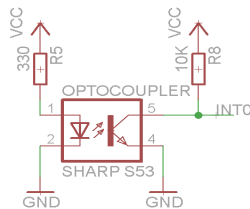
Ket: L=Low; H=High; X=Don't care

3.3.4 Perancangan Sensor Kecepatan *Rotary Encoder*

Sensor putaran atau kecepatan ini dibuat dengan sebuah optocoupler tipe “U” dan sebuah roda cacah. Sensor putaran atau kecepatan ini membaca putaran motor DC yang telah dipasang sebuah roda cacah pada poros putarnya. Sensor putaran atau kecepatan ini dibuat dengan optocoupler tipe “U” yang ditengahnya diletakan sebuah roda cacah. Optocoupler merupakan komponen optoisolator yang memiliki karakteristik penerima (photo transistor) akan mengalami perubahan logika bila terjadi perubahan intensitas cahaya yang dipancarkan oleh pemancar (LED infra merah) untuk penerima. Kecerahan led berbanding lurus dengan arus diodanya. Arus kolektor sebanding dengan tingkat kecerahan dari led sehingga dapat dikatakan bahwa arus dioda mengendalikan arus kolektor seperti transistor pada umumnya. Biasanya arus yang diperbolehkan mengalir pada infra merah adalah berkisar pada 15 hingga 25 mA. Foto transistor merupakan jenis transistor yang peka terhadap cahaya infra merah.

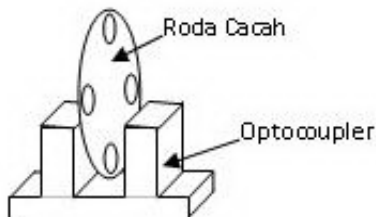


Gambar 3.10 Optocoupler “U” Sharp S53 [11]



Gambar 3.11 Rangkaian *Rotary Encoder* Dengan Optocoupler “U” S53

Roda cacah yang diletakkan di tengah optocoupler tersebut berfungsi untuk mempengaruhi intensitas cahaya yang diberikan oleh LED pada optocoupler ke photo transistor yang akan memberikan perubahan level logika sesuai dengan putaran roda cacah. Kecepatan perubahan logika photo transistor akan sebanding dengan kecepatan putaran roda cacah. Konstruksi sensor putaran dapat dilihat pada Gambar 3.12.



Gambar 3.12 Kontruksi Sensor Kecepatan

3.3.5 Perancangan Sensor Jarak

Perancangan sensor jarak ini menggunakan sensor ultrasonic tipe HC-SRF04. HC-SRF04 adalah sensor non-kontak pengukur jarak menggunakan ultrasonik. Prinsip kerja sensor ini adalah transmitter mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek. Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan obyek, sehingga jarak sensor dengan obyek dapat ditentukan persamaan

$$\text{jarak} = \text{kecepatan_suara} \times \text{waktu_pantul} / 2 \quad (3.1)$$

SRF04 dapat mengukur jarak dalam rentang antara 3 cm – 3 m dengan *output* panjang pulsa yang sebanding dengan jarak obyek. Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu TRIGGER dan ECHO. Untuk mengaktifkan SRF04 mikrokontroler mengirimkan pulsa positif melalui pin TRIGGER minimal 10 us, selanjutnya SRF04 akan mengirimkan pulsa positif melalui pin ECHO selama 100 us hingga 18 ms, yang sebanding dengan jarak obyek.



Gambar 3.13 Ultrasonic HC-SRF04

Berikut konfigurasi pin dari sensor ultrasonic HC-SRF04 seperti pada Tabel 3.3 berikut

Tabel 3.3 Konfigurasi Pin Sensor Ultrasonik

Pin	Fungsi
1	Vcc
2	<i>Trigger</i>
3	<i>Echo</i>
4	GND

3.3.6 Perancangan RFID

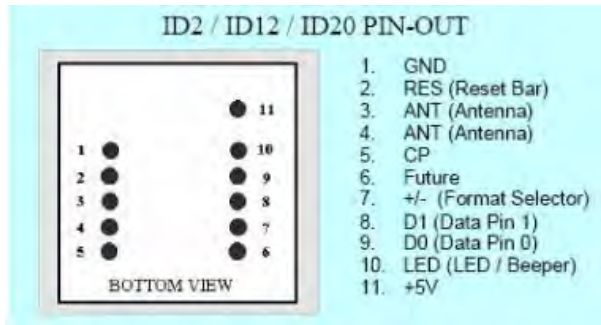
Pada pembacaan RFID Tags ini menggunakan RFID tipe ID-12. RFID ID-12 merupakan reader yang khusus mendeteksi RFID tag frekuensi 125kHz. RFID tag yang kompatibel dengan ID-12 diantaranya GK4001 dan EM4001. Dengan membaca sekitar ± 12 cm. Bentuk fisik ID-12 yang sering dijumpai diperlihatkan pada Gambar 3.14. ID12 tidak memiliki kemampuan untuk baca-tulis (Read - Write) pada sebuah *tag*. Format data yang dihasilkan oleh ID-12 berupa ASCII dan Wiegand26. Spesifikasi lengkap Modul RFID reader ID-12 dapat dilihat pada Tabel 3.3.



Gambar 3.14 RFID ID-12 [12]

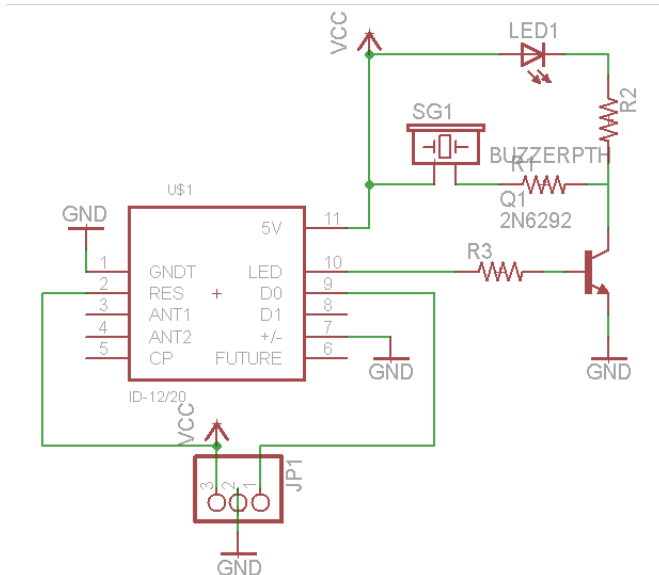
Tabel 3.4 Spesifikasi modul RFID reader ID-12 [12]

Parameter	ID-12
Jarak Baca	Sampai 12 cm
Dimensi	26mm x 25mm x 7mm
Frekuensi	125kHz
Format Kartu	GK4001/EM 4001 atau yang compatible
Encoding Manchester	64-bit, modulus 64
Jenis Catudaya	5VDC pada 30mA nominal
Jangkauan Catudaya	+4.6V-5.4V



Gambar 3.15 Spesifikasi pin pada ID-2, ID-12, dan ID-20

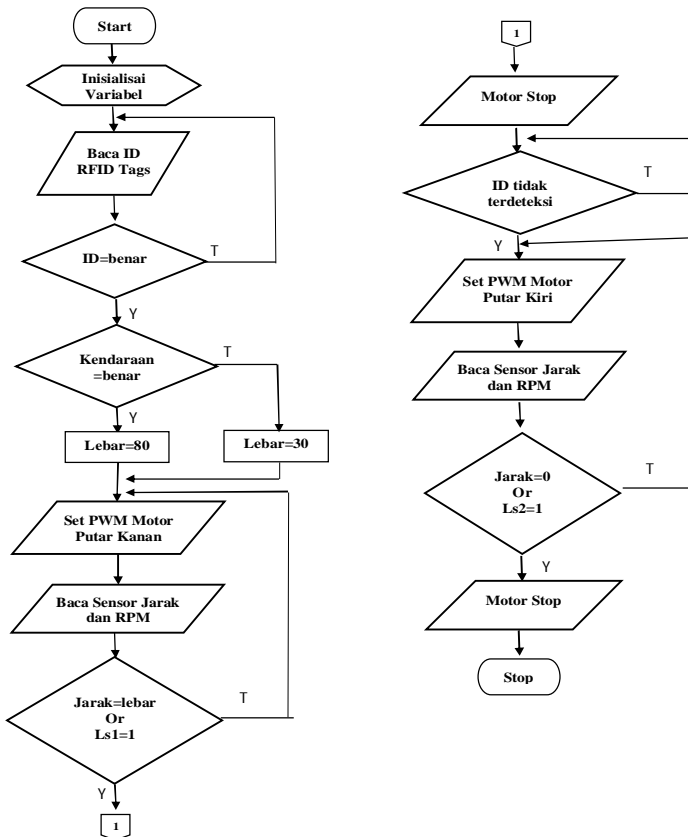
Dari spesifikasi diatas maka rangkaian yang dibuat untuk menggunakan RFID reader ID-12 seperti tampak pada Gambar 3.16



Gambar 3.16 Rangkaian RFID Reader Menggunakan ID-12

3.4 Perangkat Lunak (*Software*)

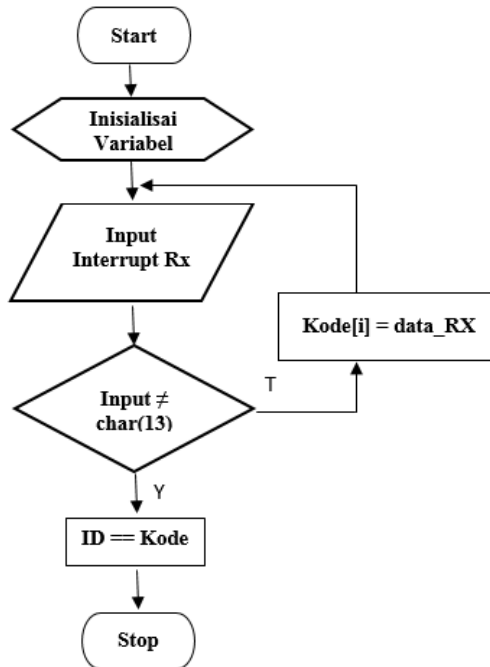
Perangkat lunak (*software*) yang digunakan dalam Tugas Akhir ini adalah CodeVision AVR sebagai compiler pada mikrokontroler AT-Mega32. Pemrograman CodeVision AVR menggunakan bahasa C sebagai pemrogramannya. *Flowchart* pemrograman sistem secara keseluruhan ditunjukkan pada Gambar 3.17.



Gambar 3.17 Flowchart Sistem Keseluruhan

3.4.1 Perancangan Pembacaan ID Pada RFID Tags

Proses perancangan pembacaan ID pada RFID Tags dilakukan dengan pembacaan data komunikasi serial dengan RFID Reader. Flowchart dapat dilihat pada Gambar 3.18.



Gambar 3.18 Flowchart Pembacaan ID pada RFID Tags

Pada ID yang dibaca RFID Tags terdapat 12 karakter Heksa dengan diawali spasi dan diakhiri enter dan pada pendeteksiannya hanya diambil 12 data karakter Heksa.

3.4.2 Perancangan Software Sensor Jarak

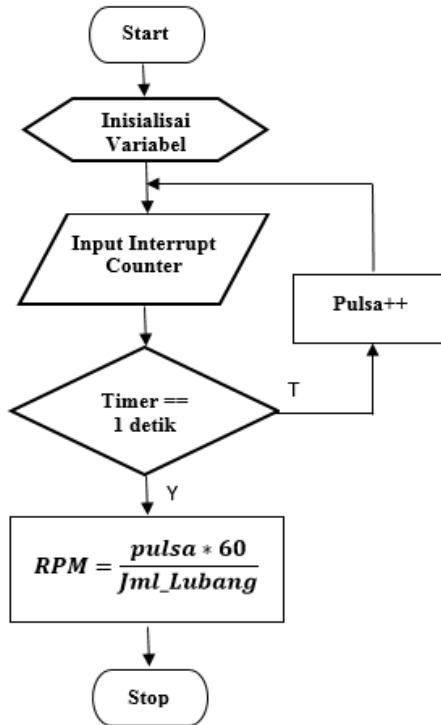
Pada proses perancangan pembacaan sensor jarak ini menggunakan ultrasonic HC-SRF04 yang bekerja berdasarkan pantulan gelombang suara yang dihasilkan oleh *transmitter* dan diterima oleh *receiver*. Inti program yang dibuat adalah fungsi `baca_srf04` yang disesuaikan dengan *timing* diagram dari datasheet.

```
    unsigned char baca_srf04()
    {
        count=0;
        //pemberian pemicu berupa pulsa positif minimal 10us utk
    aktivasi sensor
        PORTC.1=0;
        delay_us(10);
        PORTAC1=1;
        delay_us(10);
        PORTC.1=0;
        //menunggu hingga ECHO = 1
        while(PINC.0==0){};
        //mencacah untuk mengukur waktu lamanya pulsa 1 dari
    ECHO
        while(PINC.0==1)
        {
            count++;
            delay_us(58);
        }
        //mengkonversikan hasil cacahan menjadi jarak dalam cm
        jrk=count;
        delay_us(10);
    }
```

3.4.3 Perancangan Software Sensor Kecepatan

Proses perancangan pembacaan kecepatan ini menggunakan optocoupler U yang ditempatkan pada sebuah piringan hitam berlubang. Perhitungan kecepatan bekerja berdasarkan jumlah pulsa input dari optocoupler U atau prinsipnya sama dengan penghitung frekuensi (*Frequency Counter*).

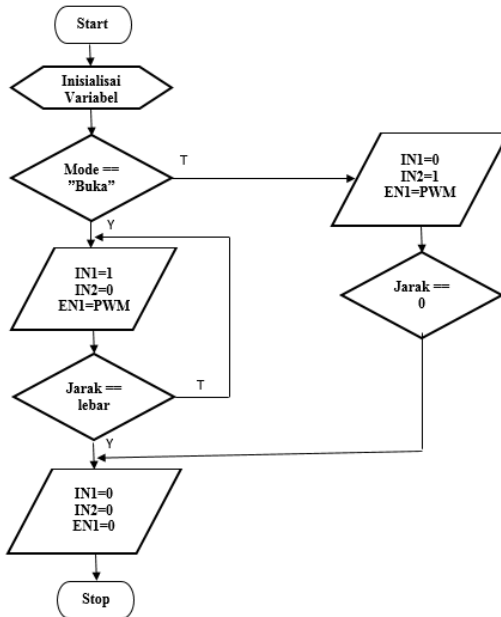
Input counter ditempatkan pada PIN INT0 (PIN D.2) dan dibaca berapa banyak pulsa dalam jangka waktu satu detik. *Flowchart* pemrogram untuk pembacaan RPM ditunjukkan pada Gambar 3.19.



Gambar 3.19 *Flowchart* Pembacaan RPM

3.4.4 Perancangan Software Pengendali Driver Motor DC

Proses perancangan pengendali *driver* motor H-Bridge mengacu pada Tabel 3.1. *Flowchart* pemrograman untuk pengendalian *driver* motor DC ditunjukkan pada Gambar 3.20.



Gambar 3.20 Flowchart Pengendalian Driver Motor DC

3.5 Identifikasi Plant

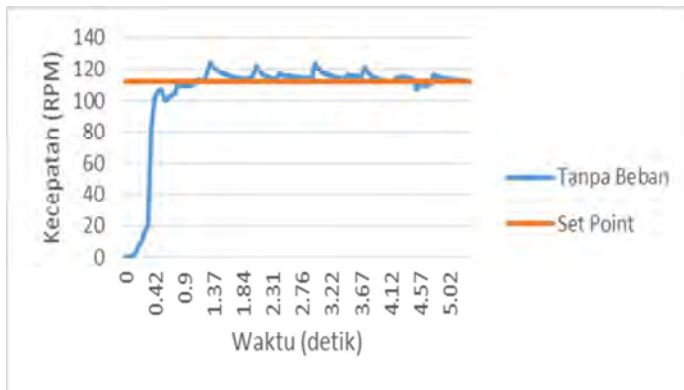
Identifikasi sistem diperlukan untuk mendapatkan respon motor secara *open loop*. Pada Tugas Akhir ini, identifikasi dilakukan adalah identifikasi statis dengan melihat keluaran respon berupa kecepatan terhadap referensi yang diberikan. Sinyal uji PWM diberikan melalui mikrokontroler agar memberikan nilai set kecepatan pada motor DC. Hasil respon *plant* yang terbaca oleh *rotary encoder* dengan *time sampling* yang digunakan sebesar 0.05 detik akan dikirim ke PC untuk dilihat grafik responnya.

3.5.1 Metode Identifikasi

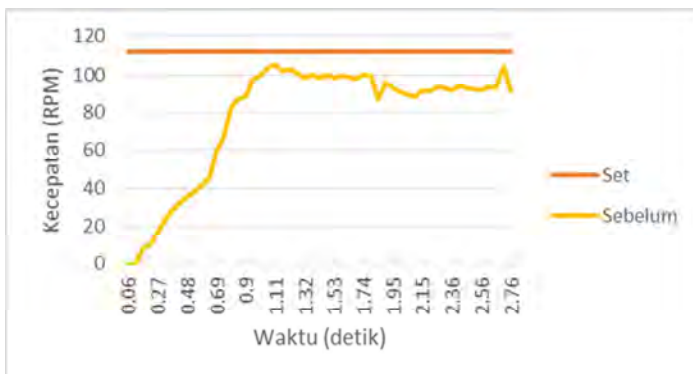
Metode identifikasi diperlukan untuk mencari respon model yang sehingga dapat dilakukan pendekatan untuk mendapatkan nilai konstanta

kontroler. Proses identifikasi dilakukan secara open loop dan statis dengan memberikan sinyal step dan melihat respon berupa kecepatan putar motor permenit (RPM). Sinyal uji step diberikan secara manual pada motor DC, kemudian hasil respon plant yang terbaca oleh *rotary encoder* dikirimkan ke komputer melalui proses akusisi data dengan mikrokontroler.

Hasil pembacaan data pada *plant* motor DC didapat respon tanpa beban dan berbeban sebagai berikut:



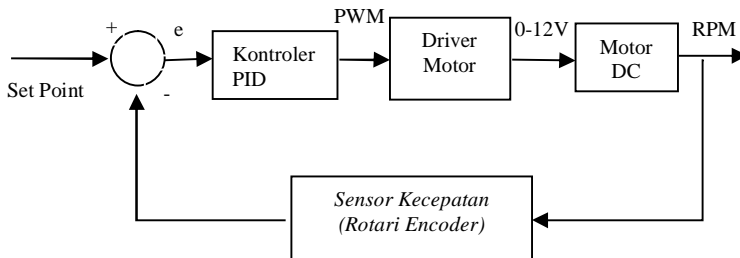
Gambar 3.21 Grafik Respon Motor DC Tanpa Beban



Gambar 3.22 Grafik Respon Motor DC Berbeban

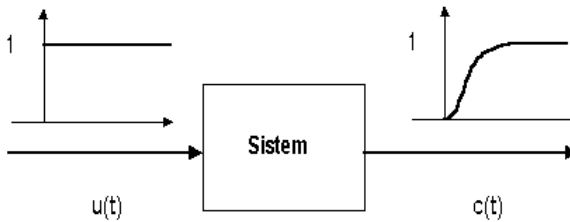
3.6 Perancangan Kontroler PID

Tugas Akhir ini akan didesain dan implementasikan kontroler PID menggunakan metode Ziegler-Nichols untuk membantu kinerja dari motor DC, sehingga saat akselerasi kecepatan yang dihasilkan sesuai dengan *set point* yang diinginkan yakni pada saat pintu awal bergerak menggunakan set point kecepatan sebesar 112 RPM, dan saat 20 cm menjelang berhenti kecepatan diturunkan menjadi 70 RPM. Sistem yang dirancang pada kontroler ini terdiri dari beberapa komponen yang terhubung satu dengan lain seperti pada Gambar 3.23.

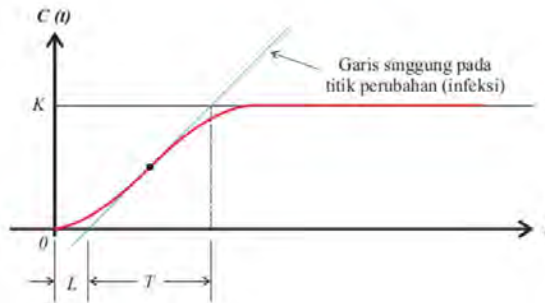


Gambar 3.23 Blok Diagram Kontrol PID

Pada tugas akhir ini menggunakan metode Ziegler-Nichols yang memiliki dua cara, metode osilasi dan kurva reaksi. Metode kurva reaksi dipilih karena model respon beban sesuai dengan syarat pada metode ini. Metode ini didasarkan terhadap reaksi sistem untai terbuka. *Plant* sebagai untai terbuka dikenai sinyal fungsi tangga satuan (Gambar 3.24). *Plant* minimal tidak mengandung unsur integrator ataupun *pole-pole* kompleks, reaksi sistem akan berbentuk S. Gambar 3.26 menunjukkan kurva berbentuk S tersebut. Kelemahan metode ini terletak pada ketidakmampuannya untuk *plant* integrator maupun *plant* yang memiliki *pole* kompleks.



Gambar 3.24 Respon tangga satuan sistem



Gambar 3.25 Kurva Respon berbentuk S.

Kurva berbentuk-s mempunyai dua konstanta, waktu tunda atau *delay time* (t_d) diwakilkan oleh nilai L dan waktu konstan atau *time constant* (τ) diwakilkan oleh nilai T [7]. Berdasarkan Gambar 3.25 terlihat bahwa kurva reaksi berubah naik, setelah selang waktu L . Waktu tunda menggambarkan perubahan kurva setelah mencapai 66% dari keadaan *steady state*-nya. Pada kurva dibuat suatu garis yang bersinggungan dengan garis kurva. Garis singgung itu akan memotong dengan sumbu absis dan garis maksimum. Perpotongan garis singgung dengan sumbu absis merupakan ukuran waktu mati, dan perpotongan dengan garis maksimum merupakan waktu tunda yang diukur dari titik waktu L .

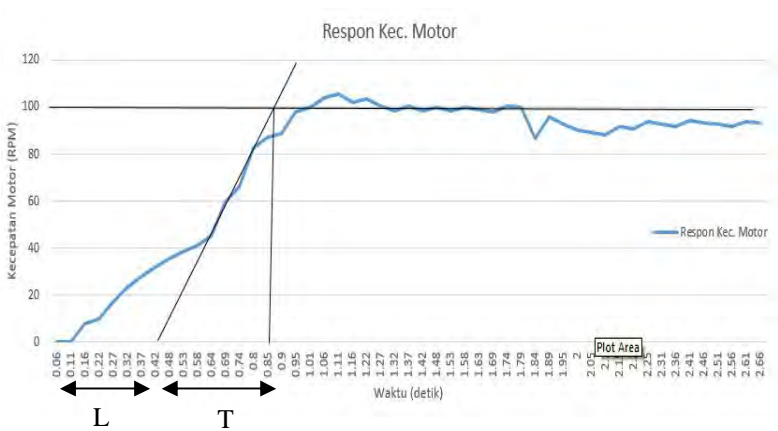
Penalaan parameter PID didasarkan perolehan kedua konstanta itu. Zeigler dan Nichols melakukan eksperimen dan menyarankan parameter

penyetelan nilai K_p , T_i , dan T_d dengan didasarkan pada kedua parameter tersebut. Tabel 3.5 merupakan rumusan penalaan parameter PID berdasarkan cara kurva reaksi.

Tabel 3.5 Penalaan paramater PID dengan metode kurva reaksi

Tipe Kontroler	K_p	T_i	T_d
P	T/L	~	0
PI	0,9 T/L	L/0.3	0
PID	1,2 T/L	2L	0,5L

Berdasarkan pada tabel tersebut bisa didapatkan nilai parameter kontroler yang diinginkan.



Gambar 3.26 Kurva Respon Berbeban

Pada respon motor berbeban tersebut ditarik garis singgung pada titik perubahan, sehingga didapatkan nilai L dan T. Nilai tersenut dimasukan kedalam tabel Ziegler-nichols untuk didapatkan konstanta PID yang dirancang. Berikut perhitungannya.

$$T = 0,44 \quad (3.14)$$

$$L = 0,42 \quad (3.15)$$

Sehingga diperoleh konstanta kontroler PID-nya:

$$\text{Untuk } K_p = 1,2 \times \frac{T}{L} = 1,2 \times \frac{0,44}{0,42} = 1,145 \quad (3.16)$$

$$\text{Untuk } T_i = 2 \times L = 2 \times 0,42 = 0,84 \quad (3.17)$$

$$K_i = \frac{K_p}{T_i} = \frac{1,145}{0,84} = 1,36 \quad (3.18)$$

$$\text{Untuk } T_d = 0,5 \times L = 0,5 \times 0,42 = 0,21 \quad (3.19)$$

$$K_d = K_p \times T_d = 1,145 \times 0,21 = 0,24 \quad (3.20)$$

Setelah didapat nilai dari masing-masing konstanta, maka kontroler PID-nya adalah sebagai berikut:

$$u(t) = K_p(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}) \quad (3.20)$$

$$u(t) = 1,145 \times (e(t) + \frac{1}{0,84} \int_0^t e(t) dt + (0,21 \times \frac{de(t)}{dt})) \quad (3.21)$$

Sehingga persamaan sinyal kontrolnya adalah

$$u(k) = (K_p \times e(k)) + (K_i \times T_s \times (e(k) + e(k-1))) + \left(K_d \times \left(\frac{e(k) - e(k-1)}{T_s} \right) \right) \quad (3.22)$$

Dimana $K_p = 1,145$

$K_i = 1,36$

$K_d = 0,21$

$T_s = 0,05$

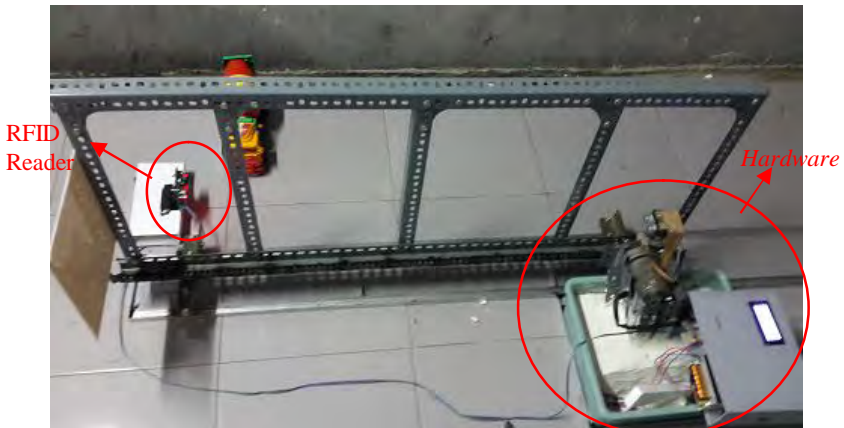
Dimana nilai Set Point kecepatan awal yang digunakan adalah 112 RPM.

BAB 4

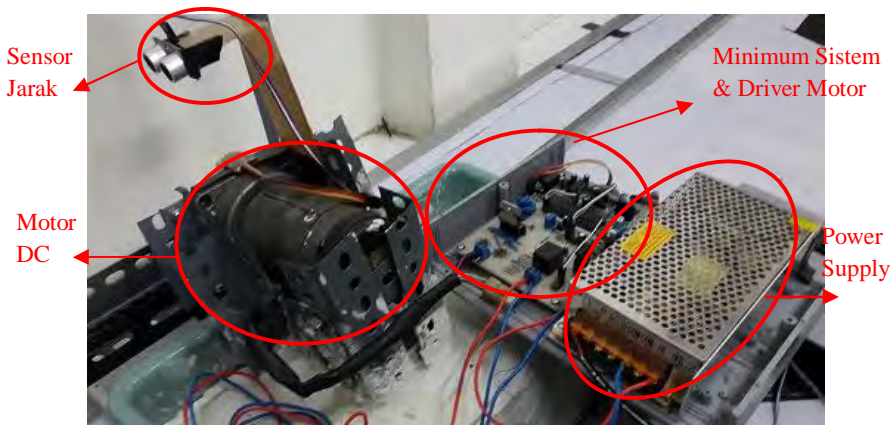
HASIL PENGUJIAN DAN ANALISA

Pengujian terhadap seluruh sistem rangkaian yang telah dibuat dilakukan setelah semua rangkaian disusun secara keseluruhan berdasarkan perencanaan pada blok diagram sistem. Pengujian dimaksudkan untuk mendapatkan evaluasi terhadap setiap rangkaian dalam sistem agar diperoleh kinerja yang lebih baik. Kinerja yang lebih baik didapatkan dengan melakukan perbaikan terhadap setiap komponen rangkaian yang mengalami kekeliruan yang diketahui saat melakukan pengujian.

Hasil integrasi secara keseluruhan dengan prototype pintu geser ditunjukkan pada gambar 4.1 berikut, sedangkan dari integrasi *hardware* ditunjukkan pada gambar 4.2.



Gambar 4.1 Integrasi Sistem Dengan *Prototype* Pintu Geser



Gambar 4.2 Integrasi Hardware

Metode pengujian dari proyek akhir ini dilakukan menjadi dua tahap, yaitu pengujian parsial dan pengujian integrasi. Dalam pengujian secara parsial dilakukan beberapa pengujian untuk rangkaian sistem minimum mikrokontroler, PWM mikrokontroler, sensor pendeteksi jarak, sensor kecepatan, pembacaan ID pada RFID Tags, dan pengujian kontroler. Sedangkan untuk pengujian integrasi, pengujian yang dilakukan meliputi keseluruhan dari sistem kerja alat berdasarkan perencanaan pada blok diagram.

4.1 Pengujian Parsial

Pengujian dikatakan baik apabila pada setiap rangkaian dapat berjalan sesuai dengan fungsi masing-masing pada plant. Pengujian dilakukan pada setiap bagian dari rangkaian sistem keseluruhan, seperti yang dijelaskan pada masing-masing sub pembahasan sebagai berikut.

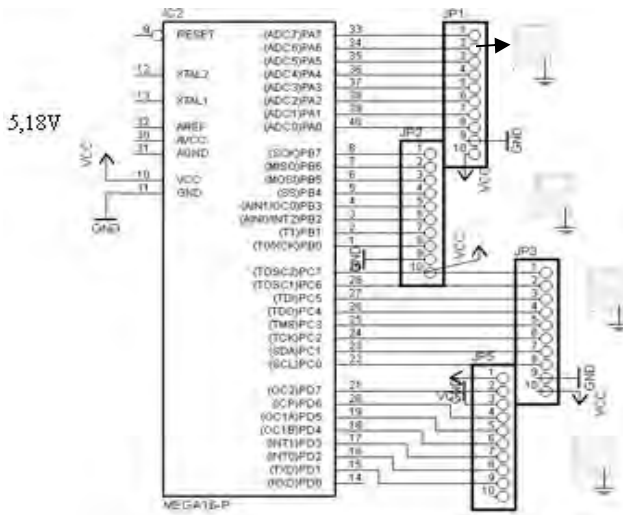
4.1.1 Pengujian Rangkaian Sistem Minimum ATmega-32

Pengujian ini digunakan untuk mengetahui bahwa *port* mikrokontroler dapat mengeluarkan nilai logika. Hal ini dilakukan untuk memastikan bahwa *port-port* yang dipakai dapat digunakan. Dari pengujian yang telah dilakukan didapatkan hasil seperti yang terlihat pada Tabel 4.1.

Untuk menandakan apakah sistem minimum tersebut bekerja, maka digunakan *LED* sebagai indikator. Dimana untuk pengujian sistem minimum langkah – langkah yang digunakan hampir sama. Berikut langkah – langkah melakukan pengujian sistem minimum (ATmega32) :

1. Membuat program menggunakan *software Codevision AVR C Compiler*
2. Membuat *project* baru. Gunakan *CodeWizardAVR* untuk menentukan *Port – Port* mana yang akan digunakan pada sistem minimum. Untuk pengisian *logic 1 (high)* maka ganti *Data Direction* menjadi *out* dan *Pullup/Output Value* menjadi 1 untuk *Port A*, *Port B*, *Port C*, dan *Port D*.
3. Setelah itu pilih opsi *File → Generate, Save and Exit* dan beri nama *file* yang sama.
4. Untuk mengecek apakah terdapat peringatan atau *error* pada program tersebut tekan *F9*.
5. Langkah berikutnya adalah men-*download* program tersebut ke sistem minimum. Terlebih dahulu, pastikan bahwa koneksi *PC* ke *chip* telah terhubung. Setelah itu tekan *Shift + F9* lalu pilih *Program to Chip*.
6. Setelah program di *download*, langkah selanjutnya adalah melakukan pengukuran masing – masing *pin* untuk tiap *Port LED* yang disertakan di *board* sistem minimum ini digunakan sebagai indikator.
7. Nyalakan *power supply*. Indikator *LED* menyala karena sistem minimum diisi dengan *logic 1 (high)*.
8. Selanjutnya adalah melakukan pengukuran di *VCC* dan di setiap *pin – pin* di *Port A*, *Port B*, *Port C*, dan *Port D* menggunakan avometer digital.
Output masing – masing *Port* dihubungkan dengan kaki positif pada avometer, dan dilihat berapa Volt tegangannya dan kaki negatif pada avometer dihubungkan pada *ground*.
9. Mencatat hasil pengukuran ke dalam tabel.
10. Untuk pengukuran dengan *logic 0 (low)*, ulangi lagi langkah 2 sampai 9. Hanya saja *Pullup / Output* diganti menjadi *logic 0*. Karena *logic 0*, maka *LED* tidak menyala.

Berikut merupakan gambar pengukuran sistem minimum pada tegangan keluaran di masing – masing *Port* mikrokontroler AT-Mega32.



Gambar 4.3 Pengukuran Tegangan Pada Masing – Masing *Port*

Hasil dari pengukuran sistem minimum I dengan *logic 1 (high)* dan *logic 0 (low)* dengan tegangan yang dihasilkan *power micro* sebesar 4,98Volt disajikan dalam Tabel 4.1

Tabel 4.1 Hasil Pengukuran Tiap *Port* Pada Sistem Minimum

PIN	PORT							
	A		B		C		D	
	High (Volt)	Low (Volt)	High (Volt)	Low (Volt)	High (Volt)	Low (Volt)	High (Volt)	Low (Volt)
0	4,45	0	4,69	0	4,74	0	4,68	0
1	4,45	0	4,69	0	4,74	0	4,68	0
4	4,45	0	4,69	0	4,74	0	4,68	0
5	4,45	0	4,70	0	4,74	0	4,68	0
6	4,45	0	4,70	0	4,74	0	4,68	0
7	4,45	0	4,70	0	4,74	0	4,68	0

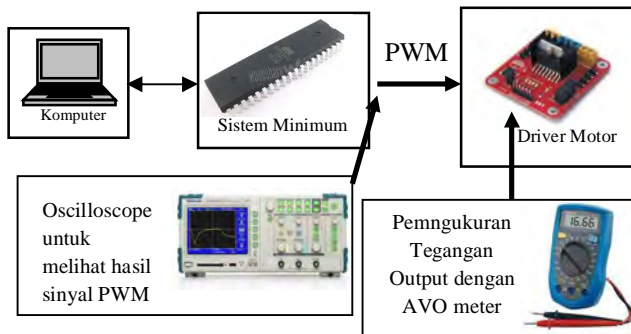
Dari hasil pengujian di tiap *Port* sistem minimum, maka didapatkan hasil bahwa *Port A*, *Port B*, *Port C* dan *Port D* terdapat tegangan apabila diberikan *active high* dan tidak terdapat tegangan

apabila diberikan *active low*. Setiap *Port* pada sistem minimum berbeda karena sistem minimum mempunyai fungsi masing – masing. Di setiap *pin* pada semua *Port* tidak terdapat masalah atau cacat pada kaki – kaki pin.

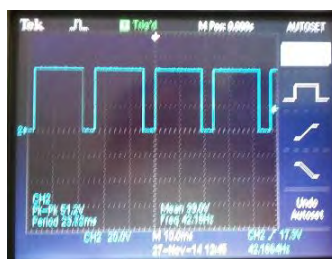
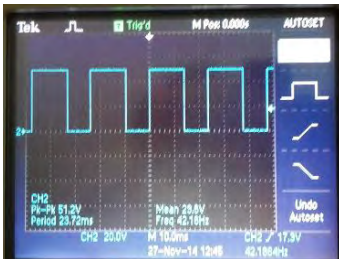
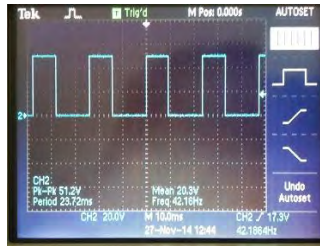
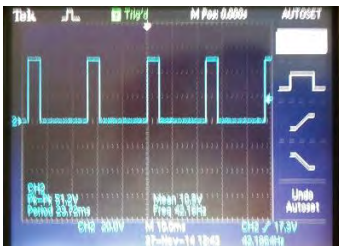
4.1.2 Pengujian PWM Dengan Driver Motor DC

Pada pengujian PWM dilakukan dengan menguji keluaran PWM dari mikrokontroler terhadap tegangan keluaran dari rangkaian driver motor DC. Pengujian ini bertujuan untuk mengetahui kemampuan PWM yang dapat digunakan, diantaranya besar nilai *duty cycle* yang dapat diatur dan menguji perubahan nilai tegangan yang dihasilkan oleh driver motor DC. Prosedur pengujiannya yakni:

1. Mikrokontroler di isikan program untuk mengeset nilai PWM 0-100%
2. Output PWM mikrokontroler pada pin OCR1A dimasukkan ke pin enable pada driver motor DC yang menggunakan L298.
3. Mikrokontroler dilakukan variasi perubahan nilai PWM 0-100% dan dilakukan pengukuran pada output driver motor DC L298.



Gambar 4.4 Mekanisme Pengujian Sinyal PWM Terhadap Output Driver Motor DC



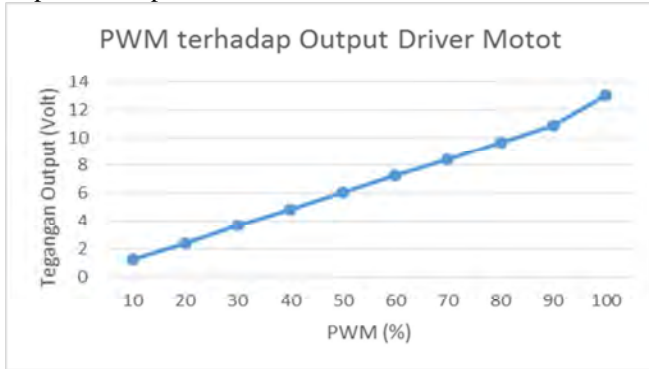
Gambar 4.5 Pengukuran Sinyal PWM (20%, 40%, 60%, 80%)

Data yang didapat dalam pengujian PWM mikrokontroler dapat dilihat pada Tabel 4.2.

Tabel 4.2 Data Pengujian PWM Mikrokontroler

Tegangan Sumber (V)	Duty Cycle (%)	Tegangan Output (V)
13,38	10	1,22
	20	2,42
	30	3,66
	40	4,85
	50	6,07
	60	7,25
	70	8,45
	80	9,64
	90	10,84
	100	12,98

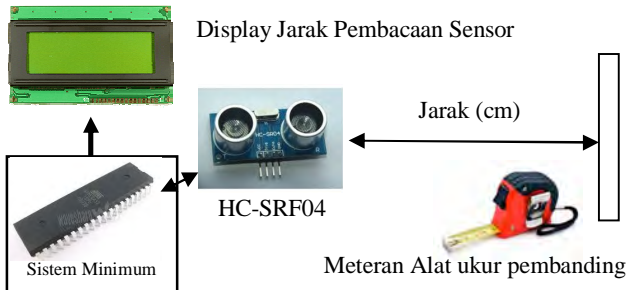
Data hasil pengujian yang dilakukan disajikan dalam bentuk grafik seperti ditampilkan oleh Gambar 4.6.



Gambar 4.6 Grafik Data Pengujian PWM Mikrokontroler Terhadap Tegangan Keluaran Driver Motor

4.1.3 Pengujian Sensor Jarak Ultrasonic HC-SRF04

Pengujian ini dilakukan untuk mengetahui hasil pembacaan dari sensor jarak sehingga dapat diketahui prosentasi kesalahan pembacaan dari sensor. Pembacaan oleh mikrokontroler dibandingkan dengan meteran konvensional untuk mengetahui prosentase kesalahan hasil pembacaan. Mekanisme Pengujian sensor jarak ini dapat dilihat pada gambar 4.7 berikut



Gambar 4.7 Mekanisme Pengujian Sensor Jarak

Hasil pembacaan sensor ultrasonic HC-SRF04 dengan meteran konvensional ditunjukkan pada Tabel 4.3

Tabel 4.3 Data Pengujian Sensor Jarak

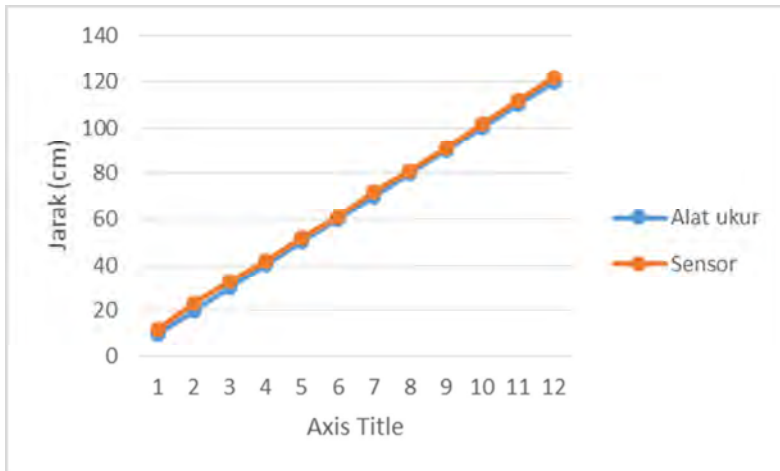
Jarak (cm)	Pembacaan Sensor (cm)	Error (%)
10	12	10
20	23	10
30	33	6,67
40	42	5
50	52	4
60	61	1,67
70	72	2,86
80	81	1,25
90	91	1,11
100	102	2
110	112	1,81
120	122	1,67
Error Rata-Rata		4,003

Prosentase error dihitung dengan menggunakan persamaan berikut:

$$error = \frac{Abs(Pembacaan Sensor - Jarak Sebenarnya)}{Jarak Sebenarnya} \times 100\%$$

Sehingga didapatkan error rata-rata dari semua pembacaan sebesar 4,003 %.

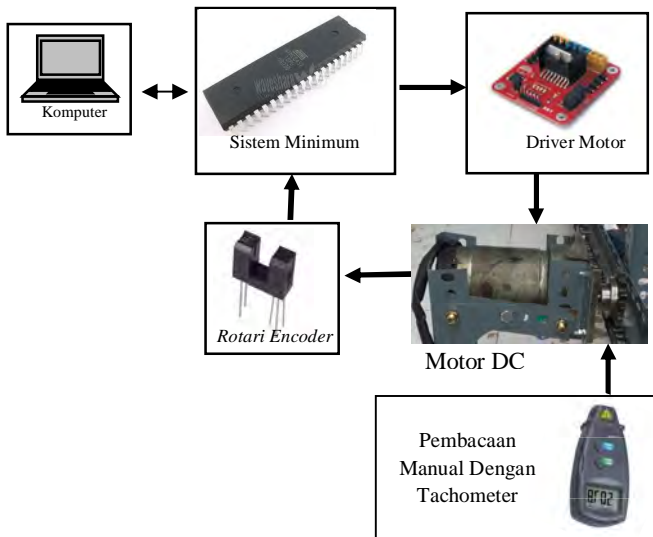
Data hasil pengujian yang dilakukan disajikan dalam bentuk grafik seperti ditampilkan oleh Gambar 4.8.



Gambar 4.8 Grafik Data Pengujian Perbandingan Sensor Jarak Dengan Alat Ukur Konvensional

4.1.4 Pengujian Pembacaan Kecepatan Motor

Pengujian ini dilakukan untuk mengetahui dan mengkalibrasi kecepatan hasil dari pembacaan sensor *rotary encoder* pada motor DC dengan kecepatan hasil pembacaan Tachometer. Pengujian dilakukan dengan menghubungkan motor DC dengan driver motor yang mana dilakukan pengubahan lebar pulsa PWM 0-100% untuk membuat motor DC berputar pada kecepatan yang berbeda. Sensor *rotary encoder* dipasang pada piringan yang terkopel dengan poros motor. Pulsa hasil pembacaan *rotary encoder* dikirim pin INT0 pada mikrokontroler yang berfungsi sebagai *input interrupt external*. Data yang didapat kemudian diolah dan hasilnya ditampilkan pada *display* yang terdapat pada mikrokontroler dan dikirim secara serial untuk dilihat grafik perubahannya. Tachometer digunakan untuk melihat perbandingan kecepatan dari piringan dengan hasil pengolahan mikrokontroler. Mekanisme pengujian kecepatan motor DC dapat dilihat pada Gambar 4.9.



Gambar 4.9 Mekanisme Pengujian Kecepatan Motor DC

Hasil pengujian untuk kecepatan motor DC dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian Kecepatan Motor DC

Duty Cycle	Kecepatan Terbaca Pada Mikrokontroler (RPM)	Pembacaan Tachometer (RPM)	Error (%)
10	63,4	61.9	2,37
20	99,8	98	1,80
30	112,7	112,5	0,18
40	118,69	116	2,27
50	121,02	120	0,84
60	123.75	121,3	1,98
70	124,08	123,9	0,14
80	125,8	125	0,64
90	126,7	126,9	0,16

Duty Cycle	Kecepatan Terbaca Pada Mikrokontroler (RPM)	Pembacaan Tachometer (RPM)	Error (%)
100	127,87	128,4	0,41
Error Rata-Rata			1,08

Prosentase error dihitung dengan menggunakan persamaan berikut:

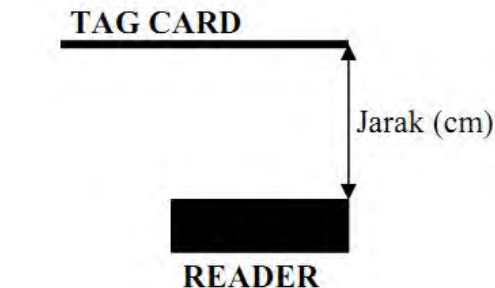
$$error = \frac{Abs(Pembacaan\ Sensor - Tachometer)}{Tachometer} \times 100\%$$

Sehingga didapatkan error rata-rata dari semua pembacaan sebesar 1,08 %.

Berdasarkan data yang didapatkan pada tabel, hasil pembacaan kecepatan motor dengan Tachometer dan perhitungan memiliki selisih yang kecil. Hal tersebut menunjukkan bahwa pengujian dan kalibrasi untuk kecepatan motor induksi telah berhasil dan sesuai.

4.1.5 Pengujian Pembacaan RFID Tags

Uji coba ini bertujuan untuk mengetahui berapa jarak pendeteksian RFID Tag Card yang dapat dilakukan oleh RFID Reader. Pengujian dilakukan dengan mendekatkan RFID Tag Card ke RFID Reader dengan jarak tertentu dan kemudian diukur oleh mistar ukur. Apabila RFID Tag Card terdeteksi oleh RFID Reader maka buzzer pada rangkaian akan berbunyi. Metode yang digunakan untuk melakukan ujicoba ini dapat dilihat pada Gambar 4.10.



Gambar 4.10 Mekanisme Pengujian RFID

Tabel 4.5 Data Jarak Deteksi RFID

Jarak (cm)	Kondisi RFID Reader	Kondisi Buzzer
12	Tidak terdeteksi	Tidak berbunyi
11	Tidak terdeteksi	Tidak berbunyi
10	Tidak terdeteksi	Tidak berbunyi
9	Terdeteksi	Berbunyi
8	Terdeteksi	Berbunyi
7	Terdeteksi	Berbunyi
6	Terdeteksi	Berbunyi
5	Terdeteksi	Berbunyi
4	Terdeteksi	Berbunyi
3	Terdeteksi	Berbunyi
2	Terdeteksi	Berbunyi
1	Terdeteksi	Berbunyi

Dari uji coba tersebut dapat disimpulkan bahwa jarak maksimal untuk RFID Reader membaca data dari RFID Tag Card adalah 10 cm. Ketika RFID Reader telah mendeteksi keberadaan RFID Tag Card maka buzzer akan berbunyi, sehingga buzzer berfungsi sebagai indikator ketika RFID Reader telah mendeteksi keberadaan RFID Tag Card.

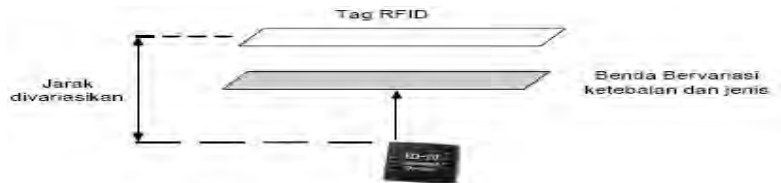
Selanjutnya RFID diuji dengan merubah posisi Tags terhadap *Reader*. Berikut hasil yang didapta pada tabel 4.5

Tabel 4.6 Pengujian *RFID* Dengan Mengubah Posisi

Posisi <i>Tag RFID</i>	Kemampuan baca
Posisi <i>tag RFID horizontal</i>	9 cm
Posisi <i>tag miring 45°</i>	4,8 cm
Posisi <i>tag RFID vertical</i>	2 cm

Selanjutnya dilakukan pengujian RFID dengan halangan. Pengujian RFID dengan halangan ini dilakukan dengan berbagai percobaan. Penghalang yang digunakan antara lain dengan kertas, tripleks, akrilik dan plat besi. Pengujian ini dilakukan untuk menentukan kemampuan *RFID reader* dalam membaca *tag* dengan penghalang tertentu. Penghalang ini dapat diibaratkan sebagai wadah atau *casing* yang paling cocok digunakan untuk box muatan pada truk. Hasil

pengukuran kemampuan baca *RFID reader* dengan penghalang adalah sebagai berikut:



Gambar 4.11 Pengujian Dengan Halangan *RFID*

Tabel 4.7 Pengujian *RFID* Dengan Penghalang

No	Jenis Penghalang	Jarak Baca
1	Kertas	9 cm
2	Triplek	6,3 cm
3	Akrilik 3mm	7 cm

4.1.6 Pengujian Kontroler PID Yang Digunakan

Pada pengujian ini mikrokontroler telah diisikan program kontroler PID terlebih dahulu. Program PID yang dimasukkan disesuaikan dengan hasil perancangan yang telah dibuat sebelumnya. Berikut hasil rancangan kontroler PI yang telah dibuat.

$$u(k) = (K_p \times e(k)) + (K_i \times T_s \times (e(k) + e(k-1))) + \left(K_d \times \left(\frac{e(k) - e(k-1)}{T_s} \right) \right)$$

Dimana $K_p = 1,145$

$K_i = 1,36$

$K_d = 0,21$

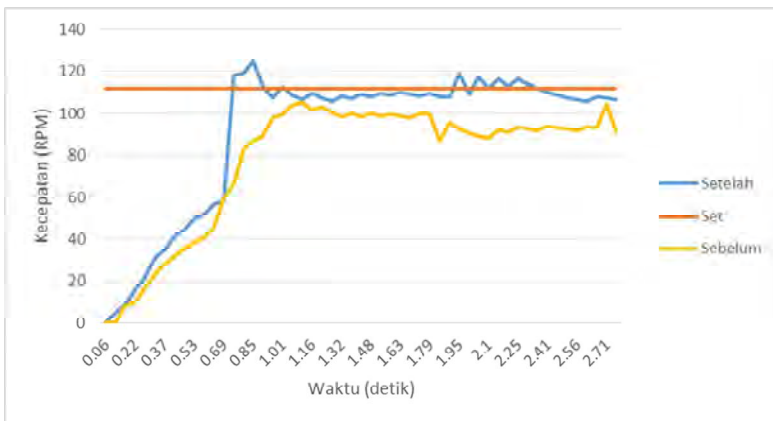
$T_s = 0,05$

$e(k)$ = error

$e(k-1)$ =error lalu

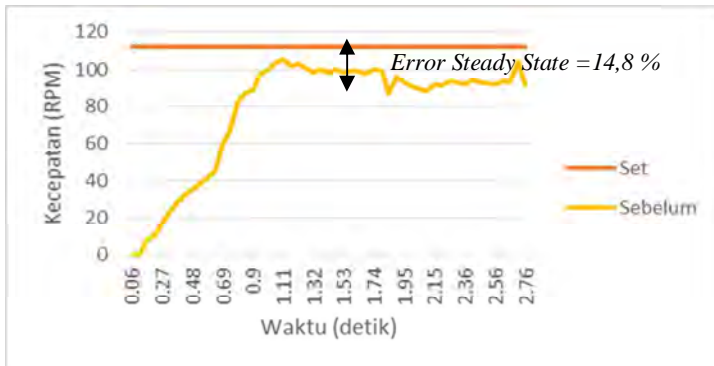
$$\begin{aligned}
 u(k) = & (1,145 \times error) \\
 & + (1,36 \times 0,05 \times (error \\
 & + error_{t_{sebelum}})) \\
 & + \left(0,21 \times \left(\frac{error - error_{t_{sebelum}}}{0,05} \right) \right)
 \end{aligned}$$

Data dari mikrokontroler akan dikirim ke komputer untuk dilihat perbandingan respon motor DC saat sebelum dikontrol dengan setelah mendapatkan kontroler. Hasil perbandingan tersebut dapat dilihat pada Gambar 4.12 berikut.



Gambar 4.12 Kurva Perbandingan Kecepatan Motor DC Tanpa kontroler Dan Kontroler PID Terhadap Set Point

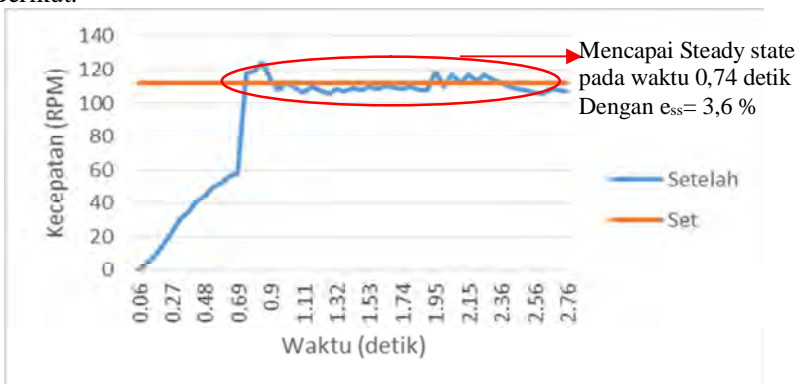
Pada pengujian tanpa kontroler terlihat respon kecepatan dari motor DC tidak mencapai nilai set point yang ditentukan. Hal ini dapat terlihat pada gambar 4.13 berikut.



Gambar 4.13 Respon motor DC sebelum dikontrol terhadap set point.

Respon motor tanpa kontroler atau secara *open loop* menghasilkan *error* terhadap nilai *set point* (*error steady state*) sebesar 14.8% sehingga perlunya ditambahkan kontroler secara *close loop* untuk memperkecil *error* tersebut.

Dari pengujian dengan kontroler terlihat terjadi perbaikan respon kecepatan motor. Dari kondisi yang tidak mencapai nilai set point, setelah dikontrol respon kecepatan motor dapat mencapai nilai set point. Hasil respon motor DC yang telah dikontrol dapat dilihat pada gambar 4.14 Berikut.



Gambar 4.14 Respon motor DC setelah dikontrol terhadap set point.

Dari hasil pengujian kontroler terlihat kurva respon setelah dikontrol dapat mencapai set point yang ditentukan dengan error rata-rata saat *steady state* 3,9% dan memerlukan waktu untuk mencapai *steady state* sebesar 0,74 detik.

4.2 Pengujian Integrasi

Pengujian integrasi yang dilakukan pada proyek akhir ini meliputi pengujian kerja sistem secara keseluruhan, yaitu pembacaan ID dari RFID Tags untuk menjalankan sistem kontrol pagar geser secara otomatis.

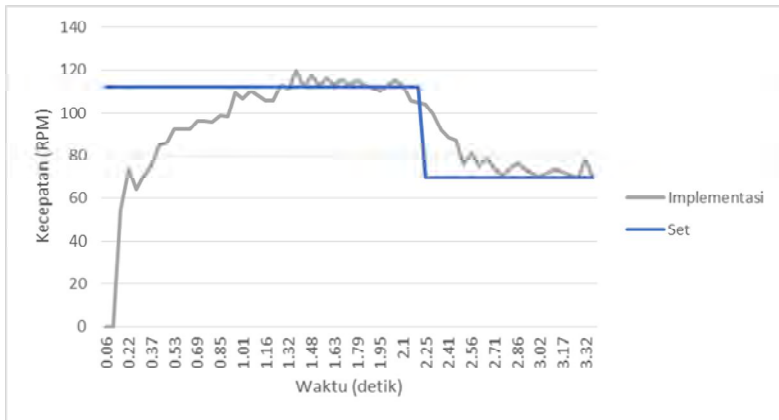
Pengujian dilakukan dengan mengidentifikasi ID untuk membuka pintu dengan lebar buka pintu yang berbeda. Hasilnya dapat dilihat pada tabel 4.8 berikut

Tabel 4.8 Pengujian Integrasi Sistem

ID Tercbaca	Identifikasi ID	Status Pembacaan ID	Hasil Pembacaan Jarak	Lama Buka (detik)
17040354BBFF	Manusia	Terbaca	42	2,86
65000850A994	Mobil	Terbaca	81	1,52
0D003F38B8B2	Manusia	Terbaca	41	3,02
0A007EA98B56	Mobil	Terbaca	83	1,25

Dari hasil pengujian terintegrasi bahwa sistem telah dapat mendeteksi ID yang telah di identifikasikan dan melakukan aksi pembukaan pintu geser sesuai jenis ID yang dimasukan.

Hasil respon Pengujian integrasi kecepatan ditunjukkan pada gambar 4.15 berikut



Gambar 4.15 Respon Kecepatan Pintu Yang berubah Saat Semakin Mendekati Batas Buka Pintu

Pengujian tersebut menunjukkan respon motor yang dilakukan perubahan nilai set point apabila jarak pagar semakin mendekati batas lebar buka.

--Halaman ini sengaja dikosongkan--

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan data hasil simulasi dan implementasi RFID Tags pada sistem kontrol pintu geser otomatis ini diperoleh beberapa kesimpulan sebagai berikut:

1. Jarak pembacaan RFID Tags menggunakan ID-12 maksimal hanya 9 cm dengan posisi Tags secara horizontal
2. Penggunaan sensor jarak ultrasonic HC-SRF04 memiliki error rata-rata sebesar 4 %.
3. Pembacaan kecepatan motor DC dengan sensor *rotary encoder* menggunakan optocoupler “U” memiliki error rata-rata 1,09%.
4. Kontrol Motor DC secara *open loop* pada *plant* pintu geser tidak mencapai set point yang diinginkan sehingga perlu ditambahkan kontroler secara *close loop*.
5. Hasil pengujian kontroler PID mampu memperkecil error rata-rata dari set point (*error steady state*) yang semula mencapai 14,8% menjadi 3,6% setelah dikontrol.
6. Pengujian Pembacaan ID pada RFID Tags dapat mengidentifikasi lebar buka pintu geser berdasarkan ID yang telah diidentifikasi yakni manusia dan mobil.

5.2 Saran

Konstruksi dari *plant* secara keseluruhan yang sederhana dan tanpa memperhatikan perhitungan secara detail dalam pembangunannya dapat mempengaruhi hasil dari kontroler yang akan didesain, sehingga hasil desain dari kontroler yang telah dibuat tidak sesuai dengan hasil implementasi.

Saran yang dapat diberikan pada penelitian selanjutnya yaitu pembacaan ID pada RFID Tags menggunakan RFID dengan jarak baca yang lebih panjang sehingga memudahkan pengguna, serta diterapkan secara langsung pada pintu geser pada rumah pribadi agar dapat diuji lebih lanjut.

--Halaman ini sengaja dikosongkan--

LAMPIRAN A

A.1 Listing Program CodeVision AVR

```
#include <mega32.h>
#include <delay.h>
// Alphanumeric LCD functions
#include <alcd.h>
// Standard Input/Output functions
#include <stdio.h>

#define echo PINC.1
#define trigger PORTC.0

// Declare your global variables here
int jarak,cek=0,lebar=60;
int mode=0,indeks=0,i,benar=0,salah=0,frek=0,tms=0,start=0;
unsigned char tempbro[22],kode[16];
unsigned char buff[20];
long int jml=1,tm=0,t=0;
//int nn=0;
//unsigned char angka[20];
//float jarak,Vjarak;
float rpm,time,sv,error,erla,uk,kp,ki,kd,out;

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
frek++;
}

#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif
```

```

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)

```

```

{
char status,data;
status=UCSRA;
data=UDR;
if(data==2) mode=1;

if(mode==1)
{
tempbro[indeks]=data;
indeks++;
}
if(data==13 && mode==1)
{
//lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("ID Terbaca");
lcd_gotoxy(0,1);
for(i=0;i<12;i++)
{
kode[i]=tempbro[i+1];
}
lcd_puts(kode);
indeks=0;sprintf(tempbro,"");delay_ms(1);
mode=2;
}
if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
// special case for receiver buffer size=256
if (++rx_counter == 0) rx_buffer_overflow=1;
#else
if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;
rx_buffer_overflow=1;
}
#endif
}
}

```

```

}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
    char data;
    while (rx_counter==0);
    data=rx_buffer[rx_rd_index++];
    #if RX_BUFFER_SIZE != 256
    if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    #endif
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;
}
#pragma used-
#endif

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Reinitialize Timer 0 value
    TCNT0=0xF5;
    // Place your code here
    if (start==1)
    {
        t++;
        if (t==1000)
        {
            t=0;
            tm++;
        }
    }
    tms++;
    //rpm=((float)frek/28)*(60/((float)tms/1000));
    rpm=(float)(frek*2142.857)/(float)tms;
    if (tms==1000)

```

```

    {
        //rpm=((float)frek/28)*60;
        frek=0;
        tms=0;
        //t++;
    }

}

// Declare your global variables here

void stop();
void buka();
void tutup();
//void keypad();
void srf();
void konv();
void lbr()
{
    if (count>=10) jbuka=(0.61*(float)count)-4;
    if (count<10) jbuka=0;
}

void main(void)
{
    PORTA=0x00;
    DDRA=0x00;
    PORTB=0x00;
    DDRB=0x00;
    PORTC=0x00;
    DDRC=0x01;
    PORTD=0x00;
    DDRD=0xE0;

    TCCR0=0x05;
    TCNT0=0xF5;
    OCR0=0x00;

    TCCR1A=0x81;
    TCCR1B=0x0D;
    TCNT1H=0x00;

```

```

TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

GICR|=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

TIMSK=0x01;

UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

ACSR=0x80;
SFIO=0x00;

ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

SPCR=0x00;

TWCR=0x00;

lcd_init(20);

// Global enable interrupts
#asm("sei")

```

```

while (1)
{
while (mode==0)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Baca ID");
if (PINC.7==0)
{mode=3; lebar=80; delay_ms(1000);}
srf();
lcd_gotoxy(0,1);
sprintf(buff,"pengahalang=%icm",jarak);
lcd_puts(buff);
delay_ms(100);      }
while (mode==2)
{
benar=0;
if (salah==0) {sprintf(buff,"17040354BBFF"); lebar=60;}
if (salah==1) {sprintf(buff,"65000850A994"); lebar=20;}
if (salah==2) {sprintf(buff,"0D003F38B8B2"); lebar=60;}
if (salah==3) {sprintf(buff,"0A007EA98B56"); lebar=20;}
for (i=0;i<12;i++)
{
if (kode[i]==buff[i])
{benar++;}
}
delay_ms(50);
if (benar<12)
{salah++;}
delay_ms(50);
if (salah==4)
{
lcd_gotoxy(0,0);
lcd_putsf("ID Tidak Terdaftar");
mode=0;
}
}
if (benar==12)
{
PORTC.2=0;
Salah=0;
mode=3;
}
}

```

```

    }
}
while (mode==3)
{
    srf();
    lbr();
    //PID Programming
    sv=112;
    if (jbuka>=lebar-15) {sv=70;}
    erla=error;
    error=sv-rpm;
    uk=(1.145*error)+(1.36*0.05*(error+erla))+(0.24*((error-
erla)/0.05));
    //konversi RPM ke PWM
    konv();
    if (jbuka<lebar)
    {
        start=1;
        buka();
    }
    if (jbuka>=lebar || PINC.6==0)
    {
        stop();
        cek=1;
        mode=4;
    }
    OCR1A=(int)out;

    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("ID Anda");
    lcd_gotoxy(0,1);
    lcd_puts(kode);
    lcd_gotoxy(0,2);
    sprintf(buff,"pengahalang=%icm",jarak);
    lcd_puts(buff);
    lcd_gotoxy(0,3);
    sprintf(buff,"RPM= %.1f",rpm);
    lcd_puts(buff);
    printf("%.1f\r\n",rpm);
    delay_ms(10);      time=((float)t/1000)+tm;

```



```

        printf("%.1f      %.1f      %.1f      %.1f      %.1f      %.1f
\r\n",sv,rpm,error,erla,uk,time);
        jml++;
        delay_ms(10);
    }
    while (mode==4)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putsf("Silahkan Masuk");
        t=0;
        tm=0;
        a1=read_adc(0);
        delay_ms(100);
        if (a1<50)
        {
            mode=5; delay_ms(4000);
        }
    }

    while (mode==5)
    {
srf();
        lbr();
        //PID Programming
        sv=115;
        if (jbuka>=lebar-15) {sv=70;}
        erla=error;
        error=sv-rpm;
        uk=(1.145*error)+(1.36*0.05*(error+erla))+(0.24*((error-
erla)/0.05))+sv;
        //konversi RPM ke PWM
        konv();
        if (lebar==80)
        {
            if (jbuka<lebar-5)
            {start=1; tutup();}
            if (jbuka>=lebar-5 )
            {stop(); cek=0; mode=0; }
        }
        if (lebar==40)

```

```

    {
        if (jbuka<lebar+5)
            {start=1; tutup();}
        if (jbuka>=lebar+5 )
            {stop(); cek=0; mode=0; }
    }
    OCR1A=(int)out;
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("ID Anda");
    lcd_gotoxy(0,1);
    lcd_puts(kode);
    lcd_gotoxy(0,2);
    sprintf(buff,"Jarak= %i cm",jarak);
    //sprintf(buff,"Jarak= %.3f cm",Vjarak);
    lcd_puts(buff);
    //lcd_gotoxy(0,3);
    sprintf(buff,"RPM= %.1f",rpm);
    lcd_puts(buff);
    delay_ms(50);
}

}

void tutup()
{
    PORTD.6=1; PORTD.7=0;
}

void buka()
{
    PORTD.6=0; PORTD.7=1; //OCR1A=76;
}

void stop()
{
    PORTD.6=0; PORTD.7=0;
}

void srf()
{
    unsigned char pulse;

```

```

pulse=0;
trigger=0;
delay_us(10);
trigger=1;
delay_us(10);
trigger=0;

atas:
if(echo == 0) goto atas;
while(echo==1)
{
    pulse++;
    delay_us(58);
    if(pulse>149){ while(echo==1);break;}
}
jarak=pulse;
delay_ms(10);

}

void konv()
{
    if (uk>=125)
        {out=255;}
    if (uk>=120 && uk<125)
        {out=(15.2*rpm)-1696;}
    if (uk>=118 && uk<120)
        {out=(13*rpm)-1432;}
    if (uk>=118 && uk<120)
        {out=(13*rpm)-1432;}
    if (uk>=112 && uk<118)
        {out=(4.17*rpm)-389.7;}
    if (uk>=98 && uk<112)
        {out=(1.86*rpm)-131;}
    if (uk>=60 && uk<98)
        {out=(0.66*rpm)-13.5;}
    if (uk<60)
        {out=(0.4333*rpm);}
}

```

--Halaman ini sengaja dikosongkan--

DAFTAR PUSTAKA

- [1] Fernanda, Oskilla. 2013 “Rancang Bangun Pintu Dan Garasi Otomatis Pada Rumah Mandiri Menggunakan Android” Tugas Akhir D4 PENS. Surabaya.
- [2] Ilyas, Muhammad. 2012 “Rancangan Penerapan Teknologi RFID Untuk Mendukung Proses Identifikasi Dokumen Dan Kendaraan Di Samsat” Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III. Yogyakarta.
- [3] Winoto, Ardi. 2010. Mikrokontroler AVR dan pemrogramannya dengan bahasa C. Bandung: Informatika.
- [4] Datasheet ATmega32/L, diakses tanggal 1 Oktober 2014, www.atmel.com/images/doc2503.pdf
- [5] Setiawan, Iwan. 2009 “Buku Ajar Sensor dan Transduser”, Universitas Diponegoro.
- [6] Datasheet HC-SRF04 Ultrasonic, diakses tanggal 1 Desember 2014, www.mikropik.com/PDF/HCSRF04.pdf
- [7] Ogata, katsuhiko, Edi Laksono (Penterjemah). 1993. Teknik Kontrol Otomatik (Sistem Pengaturan) Jilid 2. Jakarta : Erlangga
- [8] Gamayanti, Nurlita “Materi Kuliah Dasar Sistem Pengaturan” Teknik Elektro – ITS, Surabaya
- [9] Eka Maulana, S.T., M.T., M.Eng. 2012. PENGATURAN PWM (Pulse Width Modulation) dengan PLC. Universitas Brawijaya.
- [10] Datasheet L298 Dual Full-Bridge Driver, diakses tanggal 20 Oktober 2014, www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000240.pdf.
- [11] Datasheet optocoupler “U” GP1S53, diakses pada tanggal 1 Desember 2014, www.sharpsma.com/webfm_send/627
- [12] Datasheet RFID Reader ID-12LA, diakses pada tanggal 1 Desember 2014, www.id-inovations.com/httpdocs/ISO11785%20OEM%20module%20series%20ID2-12-20.pdf

RIWAYAT HIDUP



Penulis Bernama lengkap Heriyanto dan lahir di Palembang pada tanggal 10 Oktober 1991. Penulis merupakan anak pertama dari pasangan Achmad Saman (alm.) dan Siyanah. Lulus dari SMA 1 Negeri Grati Pasuruan, pada tahun 2009, penulis melanjutkan studi di Program Studi Diploma III Institut Teknologi Sepuluh Nopember (ITS), Jurusan Teknik Elektro Bidang Studi Komputer Kontrol. Pada tahun 2012 penulis melanjutkan studi di program lintas jalur Institut Teknologi Sepuluh Nopember (ITS), Jurusan Teknik Elektro dengan Bidang Studi Teknik Elektronika. Selama menjalani studi di Institut Teknologi Sepuluh Nopember (ITS), penulis di berbagai organisasi dan perlombaan. Antara lain BEM ITS, JMMI, dan HIMAD3TEKTRO ITS. Dalam prestasi penulis pernah mendapatkan penghargaan antara lain medali perunggu PIMNAS XXIV Makassar 2011, Juara 1 LCEN bidang TELEMATIKA, dan FINALIS 5 Besar dalam HARTEKNAS yang diselenggarakan oleh Kementerian RISTEK RI. Penulis mengikuti seminar dan ujian Tugas Akhir di Bidang Studi Teknik Elektronika, Jurusan Teknik Elektro, ITS Surabaya sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro. Email penulis adalah hry.achmad@gmail.com. Nomer telepon seluler 085749406709.

--Halaman ini sengaja dikosongkan--